On the Design and Development of Musculoskeletal Bipedal Robots

TECHNISCHE UNIVERSITÄT DARMSTADT

Vom Fachbereich Informatik der
Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

Dissertation

von

Dipl.-Inform. Dorian Scholz
(geboren in Frankfurt am Main)

Referent:	Prof. Dr. Oskar von Stryk
Korreferent:	Prof. Dr. André Seyfarth
	(Institut für Sportwissenschaft, Technische Universität Darmstadt)
Tag der Einreichung:	21.07.2015
Tag der mündlichen Prüfung:	18.08.2015

D17
Darmstadt 2016

Contents

List of Figures

List of Tables

1 Introduction and Motivation

1.1 Motivation

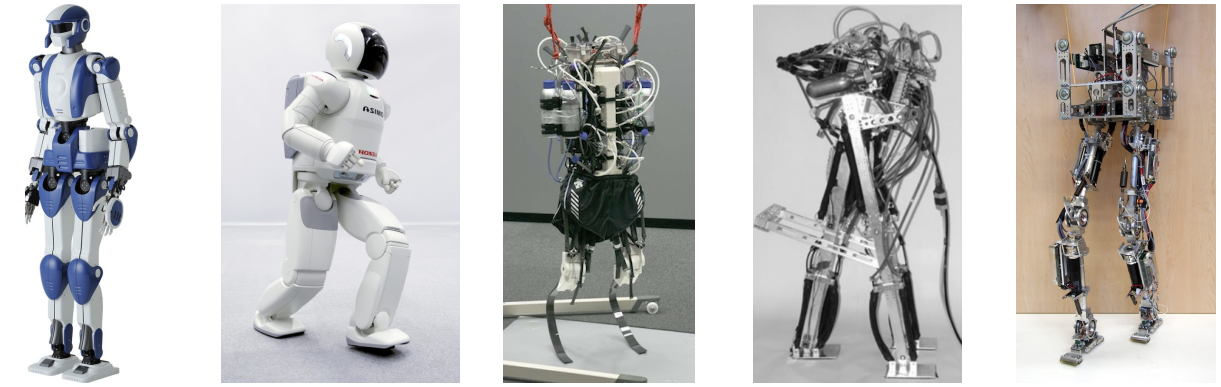
Even though bipedal walking and running are tasks which are solved by humans every day, achieving a similar performance with a robotic system is difficult. While current bipedal robots can perform walking and slow jogging motions, they cannot for example run a marathon like humans can. To gain efficient, robust and versatile locomotion for bipedal robots a diverse set of scientific and technological problems need to be solved. Bipedal locomotion, especially running, includes impacts of the feet on the ground leading to impact shocks in all leg joints. This requires high robustness and good shock absorption of the mechanics as well as appropriate trajectories to ease these impacts. But many of today's bipedal robots are still using a rigid coupling between actuators and joints [21, 30] (e.g. Figure 1.1 (a) and (b)), which gives them good control over the joint trajectories, but also makes the actuators subject to these impact shocks. Furthermore, to increase efficiency, the energy of the impact forces need to be transferred into propulsion for the next step. Both, mechanical robustness and energy efficiency, can be improved in principle using mechanical elasticity as can be seen in biological legged systems [1, 3, 37]. To cope with the additional complexity and challenges introduced by the elasticity, the mechanical design and the control need to be highly adapted to it. Furthermore, a mechanical design that not only includes joint elasticity, but rather a musculoskeletal design with additional biarticular structures spanning over two joints, can significantly improve performance [7] if properly designed. The resulting passive dynamics can aid motion control. However, proper design is crucial and is a challenging task. To leverage the potential of a combined passive mechanical and active digital control, both of them need to be carefully developed and optimized together systematically to deliver the desired output performance of the robot. Only then will it be possible to achieve the highly challenging task of performing efficient, versatile and fast locomotion with robust postural stability with a bipedal humanoid robot.

1.2 Goals of this Work

To advance the state of dynamic locomotion with musculoskeletal bipedal robots work has been done in multiple areas.

In Chapter 2 of this thesis an overview of the state of research in fields relevant to this work is given outlining the design concepts as well as mechanical considerations for the development of musculoskeletal robots.

Next the requirements on the system and software architecture needed for the control and monitoring of a prototype series of musculoskeletal robots are derived in Chapter 3. Here it is found that a prototype series of musculoskeletal robots requires a different approach on



(a) HRP-4 **Source:** KAWADA, Inc.

(b) ASIMO **Source:** Honda, Ltd.

(c) Athlete **Source:** [47]

(d) Hosoda's biped **Source:** [23]

(e) BioBiped2 **Source:** own photo

Figure 1.1: Examples of conventional (a, b) and musculoskeletal (c, d, e) bipedal robots.

system and software architecture design compared to what is used for conventional legged robots. Since the musculoskeletal leg design based on tendon driven series elastic actuators [52] with additional biarticular structures is much more complex than for conventional rigid robots, common approaches to design and control are not well applicable. Also, in a prototype series the robots' specifications will be changing over multiple generations. These include fundamental properties such as the number of joints and actuators as well as types and number of sensors. Furthermore, the amount of sensors is larger to allow collecting more comprehensive data about the robot's motions. These data are used to analyze its behavior and lead to design decisions for the next prototype generation. The control system architecture must be flexible to enable investigation of an unusually large variety of different control methods for research purposes (ranging from conventional cascades of single-variable to novel multi-variable feedback as well as to novel feed-forward control concepts). Because only controlling all actuators at full control rate with knowledge of all low-level sensor data allows accounting for the dynamic interactions of the different links additionally coupled by biarticular structures. As this robot prototype series also functions as a testbed for hypotheses from biomechanics the control system has to be flexible enough to allow the implementation of different control approaches, including biological inspired control concepts.

In Chapter 4 the design decisions for the BioBiped robot series' hardware and software components based on the aforementioned requirements are described. It further discusses the reasoning behind changes made between the different generations of the robot in the evolution of the prototype series based on results from the systematic evaluation of basic functionalities of the robot described in Chapter 6. Also, an overview of the implementation of the software components is given, including the technologies used and the newly developed development tools published as open source.

Chapter 5 evaluates different control concepts with respect to their applicability to the musculoskeletal BioBiped robots. On the gait level, control is investigated based on state machines realizing different motions with state transitions based on time or sensor input data while on the joint level values measured by different sensor inputs can be used as con-

trolled variable. The concepts include standard control approaches like PD motor or joint position control, novel feed-forward generated motor voltage trajectories based on learned models and combinations of both. With the combined feedback and feed-forward control a biologically inspired control with a high latency feedback component is also investigated.

To show the validity of the mechatronical design of the robot, the software architecture and the control concepts developed in this work, systematic experiments were performed with two different robot prototypes in Chapter 6. These experiments are designed to systematically evaluate the robots' basic functionalities to allow for improvement of the current and next robot generations. The first experiments investigate the basic motion abilities of the robots' mechanical design and its actuation system with respect to performing dynamic bouncy motions, which is a requirement to achieve the goal of jogging. Further, it is investigated if the robots are capable of thrusting themselves off the ground and maintaining a bouncing gait over time, even when externally perturbed.

In Chapter 7 a new concept for implementing hardware-in-the-loop motion optimization of motion relevant parameters for bipedal musculoskeletal robots is presented. So far the fine-tuning of parameters of the design of the mechanics and the control system of musculoskeletal robots has been mainly based on trial and error experiments with the robot hardware [23, 47]. This is caused by the difficulties of the development of sufficiently accurate multibody system (MBS) models for simulation of dynamic motions of a musculoskeletal robot. Even though a highly advanced MBS dynamics simulation model has been developed for the BioBiped robots used in this work [61], it is not enough to optimize only in simulation. The still limited ability of such models to sufficiently accurately represent all relevant properties and aspects of the real robot prototype and the resulting gap between simulation and real motion behavior prevent direct transfer of the results to the robot (see for example differences in Figure 7.3 and further details in Section 7.1). Further, the hardware of a robot prototype changes over time through new mechanical developments, but also due to wear of the mechanics during operation. So keeping the simulation model updated with these changes would be required in addition to constantly repeating the tedious process of system identification.

On the other hand, using only trial and error for the design of mechanics and control system for a musculoskeletal robot will not lead to an optimal solution due to the complexity of the system. With established optimization methods in a hardware-in-the-loop setup, it is possible to determine a reasonably good solution, but the number of robot experiments this requires makes it practically infeasible.

Therefore, the new concept described in Chapter 7 uses the simulation model only as means to reduce the number of robot experiments needed during the hardware-in-the-loop optimization. This is achieved using an expert to interpret the simulation results based on his biomechanical understanding of the robot to help guide the parameter optimization process.

The thesis closes with a conclusion in Chapter 8.

2 Related State of the Art in Bipedal Robots

Many of the bipedal robots that exist today are designed as rigid kinematic chains with stiff actuation systems [21, 30] in contrast to biological systems, which are powered by highly elastic muscle-tendon actuation units. Stiff rotary joint actuators are often preferred as they can be controlled using well-known and proven control strategies for precise motions and high repeatability. Based on these motor control concepts, control strategies for dynamic postural stability (e.g. ZMP [84]) with high frequency closed-loop controllers were developed using joint position, torque and contact forces as feedback. This approach led to legged robots able to perform dynamically stable walking and even slow jogging motions in structured, well-defined environments. But these robots without mechanical compliance and elasticity can be damaged through unplanned contact with the environment and have no means of storing impact energy from ground contacts and releasing it in the next gait cycle. In order to achieve more robust locomotion on uneven ground and to develop faster running motions with this conventional approach, even higher control rates and power output would be needed.

In the human locomotor system these requirements are solved using compliant and elastic muscle-tendon complexes as actuators. Implementing these in a robotic system through series elastic actuators leads to a joint-elastic robot (e.g. M2V2 [53]). This adds passive protection of the motors and gears from impacts [33] and bears the potential to store energy in the elastic elements. But the added elasticity also has unwanted effects like oscillations, possible joint over-extension and reduced precision in joint position tracking. Due to the oscillations introduced by the added elasticity, the complexity of the control system necessary for such joint-elastic robots is highly increased. While high precision and repeatability as required for stiff robot arms is not needed for running and walking, it requires fast synchronization of joint motions in a robust manner and also being able to compensate for disturbances (like from uneven ground).

One element of the solution to reduce the complexity needed in the control system can also be observed in the human locomotor system. Besides the mono-articular muscles attached to each joint, additional biarticular muscles spanning over two joints are found in the human leg [26, 88]. They help to synchronize joint movements and distribute power between joints [15, 28, 83]. These functionalities can be implemented in a robotic system as active or passive biarticular elastic elements leading to a musculoskeletal robot design. A musculoskeletal robot can potentially use these positive effects of the biarticular structures, but needs to be carefully designed and controlled in order to do so. If not setup properly the addition of biarticular structures, which leads to an even more complex mechanical system with multi-joint coupling, needs an even more complex control strategy. So great care has to go into the design and setup of these structures to actually make use of their positive potential. A new approach to the efficient setup of these structures is investigated in Section 7.3.

2.1 Bipedal Robot Actuation and Locomotion

Early research on bipedal robots was done in Japan at Waseda University with the WL and WABIAN robot series [35], producing the first humanoid robot to perform dynamically stable walking based on zero-moment point (ZMP, [84]).

Prominent developments took place at the Honda Motor Company leading to the well know ASIMO robot series [20, 21, 76]. Recently, dynamically stable autonomous walking robots were presented, which can jog with a small flight phase [21, 29] or hop on one leg. Nevertheless, these models have rigid actuation without any intentional mechanical elasticity and walk with flexed knees. This leads to high energy consumption and very little energy transfer between strides. Other well-known examples of rigid fully actuated robots based on ZMP are Johnnie [13] and the HRP series [30].

A different approach is used in so-called passive dynamic walkers, introduced by McGeer [44]. Through appropriate mechanical design they can walk dynamically stable down a slope without active control or additional energy input. Variants with active energy input exist, which can walk on flat ground. While performing their motion very energy efficient, they have a very limited area of stability and lack the versatility known from human locomotion.

A series of dynamically stable hopping robots was developed at the MIT Leg Lab by Raibert and his team [63]. They maintain their stability by controlling the leg landing angle and leg thrust for each ground contact. Although they are stable at highly dynamic motions, these robots have elastic telescopic legs without feet, making it impossible for them to stand still. A state machine is used to switch between different control outputs for the legs' actuators depending on the current phase of the gait cycle.

The robot MABEL also features elasticity, but on a two-segmented leg without feet [14]. Here the elasticity is implemented using leaf springs and a wire transmission to couple hip and knee joints. This allows one actuator to control the leg length while another one controls the leg angle, making it similar to Raibert's hopping robots from a control perspective. While MABEL's control system also utilizes a state machine, it is not used for trajectory generation, but rather for transitions between different controllers for different states of instability [50].

2.2 Musculoskeletal Bipedal Robots: Design and Parameter Optimization

Only a few musculoskeletal bipedal robots designs have been investigated so far.

Hosoda and his team developed and evaluated musculoskeletal robot leg designs with pneumatic actuation (for example Figure 1.1(d)). The bipedal design described in [23] was able to perform feed-forward based walking, jumping and a short sequence of running steps using antagonistic mono-articular actuators. For each motion performed, the motion parameters were set up using manual tuning without utilizing a simulation model or an optimization method.

In [22] the development of a monopod is described, which is activated using biarticular structures passively for joint coordination and mono-articular muscles to induce power.

A hopping motion was generated using a central pattern generator to activate the mono-articular muscles with feed-forward control based on observation of human motion. No simulation model was used according to the publications, but rather direct tuning of parameters on the robot. This was performed in a manual trial and error fashion based only on data from an external motion capture system, no internal sensor data of the robot was used.

A new bipedal robot is presented in [38] where it is used to evaluate the improvement of the robustness of a hopping motion through muscle reflexes. Hundreds of robot experiments were needed to evaluated the effectiveness of the reflex, as again no simulation model was used.

Niiyama et al. also developed pneumatic bipedal musculoskeletal robots in [47, 48]. A simulation model was used in [48] to adapt human muscle activation patterns to fit the Athlete robot (Figure 1.1(c)). But still the motion was manually tuned on the robot afterwards.

In [75] the highly underactuated musculoskeletal JenaWalker II robot (Figure 4.1(a)) is described, with three-segmented legs powered by a single DC-motor at the hip using sinusoidal patterns. It uses additional servo motors to adjust the rest lengths of its passive biarticular structures allowing it to change the leg posture. The knee and ankle joints are actuated through passive couplings with the hip using wires and springs. It was shown that the robot is able to perform walking and jogging motions on a treadmill, while being externally restricted to movement in the sagittal plane. To achieve this, manual tuning of the parameters of the passive structures was needed.

The insights gained from the JenaWalker robots are the basis for the design of the BioBiped robot series used in this work. The BioBiped robots also feature a musculoskeletal design, but fully actuated with at least one motor per joint. The tendon driven series elastic actuation also uses DC-motors, which are more precise to control and easier to monitor than pneumatic actuators. Further, the BioBiped robots feature a large number of internal sensors for control and also analysis of the motions as described in Chapter 4.

While a simulation model was used in the design phase of the BioBiped1 robot for the actuator dimensioning [62], the first motion experiments on the robot described in [60] still used manual selection for adjustable hardware parameters, like spring stiffnesses and lever arm lengths. Since then a detailed multibody system dynamics simulation was developed for the BioBiped robot series in [61]. But a direct transfer of simulation results to the robot is still not possible due to the limited ability of such models to represent the real robot sufficiently accurate with all relevant properties. As the number of parameters of musculoskeletal robots is higher than for conventional rigid robots, a hardware-in-the-loop parameter optimization on the robot would be expensive with respect to time needed for experiments and also wear on the robot prototypes. Therefore, an approach is needed to systematically use knowledge gained in the simulation to make the parameter optimization on the robot more efficient. This thesis presents such a systematic approach to parameter optimization on the hardware for musculoskeletal robots in Section 7.3.

2.3 Evaluation of Basic Motion Functionalities of Musculoskeletal Robots

During development of a new robotic system it is important to test and evaluate its basic functionalities like mechanical robustness or the actuator system dimensioning. In case of elastic musculoskeletal bipedal robots also parameters like the passive energy restitution through the elastic structures are relevant. Unfortunately publications about the evaluation of basic functionalities for musculoskeletal robots are rare. Only for the Athlete robot an experimental evaluation of the mechanical robustness against a drop from a height of 1 m is described in [47]. But the robot is in a configuration, where it only damps the fall and does not rebound off the ground. So no data for energy restitution or ground reaction forces (GRFs) are given.

In this thesis a systematic approach for the evaluation of basic functionalities for the musculoskeletal BioBiped robots is presented and its results are described in Chapter 6.

2.4 Electronic Control System Architecture for Bipedal Robots

The most common approach when designing electronic control systems for robots is a hierarchical control architecture which uses micro controller units (MCUs) for individual motor control and sensor processing and connects them via a bus system to a central control system responsible for higher level tasks like set point or trajectory generation. Usually each MCU controls a single joint motor using position, speed or torque control based on a set point given by the central control system. The controlled variable is either read directly from a sensor or derived from processed sensor data on this MCU. This allows the motor control loop to run locally at a high rate and with hard real-time constraints on the MCU, while the central control system updates the set point on a less frequent basis via the bus system. While this type of hierarchical control relaxes the real-time constraints on the central control system and does not require a high bandwidth nor a low latency bus system for the set point updates, it does not allow for full multi-variable control, as each MCU only knows about the sensor data from its own motor and joint.

Still, this approach is used in many current bipedal robots as for example Johnnie [39], where on a computer desired position or torque trajectories for the joints are generated at a rate of 250 Hz. Distributed micro controllers implement the control for the motors at a control rate of 2.5 KHz using feedback data only from their directly connected joint position or force sensor.

In many conventional rigid bipedal robots, digital servo motors are used. They already include the MCU with an integrated feedback controller logic and offer only a low bandwidth bus system interface which does not allow for an external central controller to directly control the motor. Examples of this type of control architecture with digital servo motors with their own MCU performing local control tasks can be found in many robots currently available [17, 72, 87].

To enable true multi-variable control as a research opportunity, which can also be model based, a new control system architecture is presented in Section 4.4. Based on this different control concepts are investigated in Chapter 5.

3 System Requirements for a Prototype Series of Elastic Robots

The system used to control and monitor a robot consists of electronic components like sensors, actuators and processing units (like micro controllers and PCs) as well as the control software. Depending on the system setup the software can be run on one or multiple of the different processing units used to process sensor data in different ways. The requirements for a control system to be used in a series of musculoskeletal robot prototypes which function as scientific testbeds are different from the control of an already developed robot performing well defined tasks. On a development platform not as much information about the setup of the robot and its desired motions is known in beforehand. Over the course of developing different hardware generations the sensor and motor system will change as well as the desired motions and the applied control concepts. Therefore, a more flexible setup of the control system is needed in order to reuse the developed components in later robot generations. Further, the primary goals for such a development system not only include the control of the robot prototypes. Also, gathering of sensor data relevant for the detection of undesired behavior and for the development process of the next robot generations is of high importance.

Sensor data are used in the following scenarios with different requirements:

- control of the robot under hard real-time constraints
- live monitoring of the robot by the operator under soft real-time constraints
- recording of data for later offline analysis

Due to the ongoing hardware development the number and type of the actual sensors and actuators is not known at the start of the system development and will change over time as will the evaluation criteria and use of the sensor data. For example from one generation to the next new joints and actuators could be introduced and need to be monitored and controlled using additional sensor data.

When using elastic actuation one also requires several times as many sensors than in rigid systems to measure the state of each actuator and joint, e.g. position sensors before and after the elastic element (compare Figure 3.2(a) to Figure 3.2(b)). Further, to allow for a more comprehensive analysis of the system's behavior during development, such a development platform typically has more sensors and generates measurements at a higher sampling rate than are actually needed for the control.

Therefore, a flexible system architecture with a high bandwidth bus system is needed to be able to quickly integrate new sensors and also change the way data are used for control. Recording of all available raw sensor data is important to be able to process and analyze them using different algorithms later on. Analyzing these data plays an important role in improving the mechanics, the control concepts and the control parameters for the current system, but

also in the planing of the next prototype generations. Processing of these raw data in real-time into physical units and calculating derived data is important for model based control approaches. Also, these converted data can be used for offline analysis e.g. in comparison with human motion data.

3.1 Requirements for Control

In conventional robots with stiff couplings of actuators and joints and a one-to-one mapping of joints to actuators typically local joint level controllers are used. One for controlling each actuator based on the sensor data from its corresponding joint (single-input single-output (SISO) shown in Figure 3.1, see also 'Independent Joint Control' in [5]). These local joint level controllers usually get set points for e.g. desired joint position, speed or torque from a gait level controller.

In a musculoskeletal robot the actuator to joint mapping is more complex due to the antagonistic setup, the use of biarticular structures and the highly elastic coupling between them. To be able to control such a system where especially during dynamic motions each joint is influenced by multiple actuators a coordinated interplay between the controllers of all involved actuators is required. For such a complex system a true multi-variable controller (multiple-input multiple-output (MIMO) shown in Figure 3.1) is needed which can control trajectories for all actuators based on the input from all sensors of all joints. Furthermore, the multi-variable controller needs some model knowledge to be able to take advantage of the system's dynamic properties. This model knowledge can be implemented as an inverse dynamics model derived through mathematical methods or it can be learned based on system identification data. While the former allows for a description of the system's behavior in its complete range of motions, it is very difficult to create a model of sufficient quality for such complex elastic robots. A learned model on the other hand can be very accurate in the motions used to teach the model, but is less reliable when extrapolating to other motions. So, a combination of derived and learned models could be a viable approach.

Besides the control of the actuators, the generation of goal trajectories for the overall leg is an essential part in the motion generation process. Even though this is also relevant for conventional robots, it is even more important for elastic robots. Their dynamics can be used to their advantage only when the planing of the trajectory is done with respect to the elastic properties of the system. With the use of an inverse dynamics model in the trajectory planing, the system can also perform motions using feed-forward control with only a small amount of feedback control for corrections. This is a similar approach to what can be seen in human motion generation, where the feedback loop via the nervous system is too slow for feedback control of fast motions. Instead, a very accurate inverse dynamics model learned over many repetitions is used to improve trajectory generation and feed-forward control of highly dynamic motions.

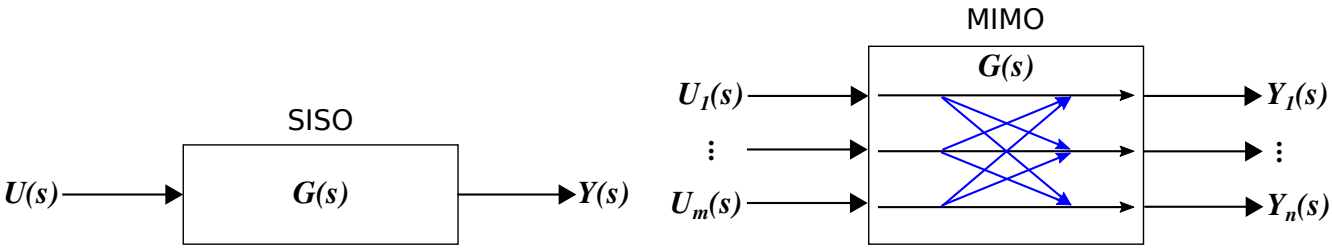


Figure 3.1: Diagram of generic single-input single-output (SISO) and multiple-input multiple-output (MIMO) controllers. In the MIMO controller, the transfer function G generates multiple outputs Y depending on multiple inputs U.

Source: own representation

3.1.1 Joint Level Control

For a complex elastic robot prototype series it is not known in advance which aspects of the control to be developed will depend on which sensor inputs. So a typically used distributed control system (DCS) with a position or torque control on a per joint or actuator basis would not allow the individual controllers to account for the dynamic interactions between multiple links. Therefore, it is advantageous to allow for real multi-variable control by implementing a centralized control system where the raw data from all low-level sensors can be used to generate joint level control outputs for all actuators, without depending on a cascade of underlying controllers. By realizing the control system in software in a central computing unit it is flexible in terms of the controller structure and allows implementing multi-variable control of all combinations of sensor input and control output. To allow such a flexible control system to be prepared for the inclusion of additional sensors and keep the controller working in real-time, a high bandwidth and low latency bus system connecting the components is required.

3.1.2 Gait Level Control

Besides the actuator-level control system, also the gait-level control needs to be able to accommodate different control concepts. Control on the gait-level can be achieved using model based approaches to control properties like the leg stiffness, leg angle and thrust generated by the leg. The use of a potentially processing power intensive model requires the control system to have enough processing power to evaluate this model in time for each control cycle. Alternatively to model based approaches an implementation of a central pattern generator, a parametrized trajectory generator or a state machine can be used for trajectory generation. While a state machine based approach has to work online using real-time sensor input, some model based approaches can also be used to generate control outputs offline for feed-forward control. To allow the evaluation of these different concepts a modular control system is needed, where they can be easily exchanged and combined.

3.2 Requirements on the Electronic System

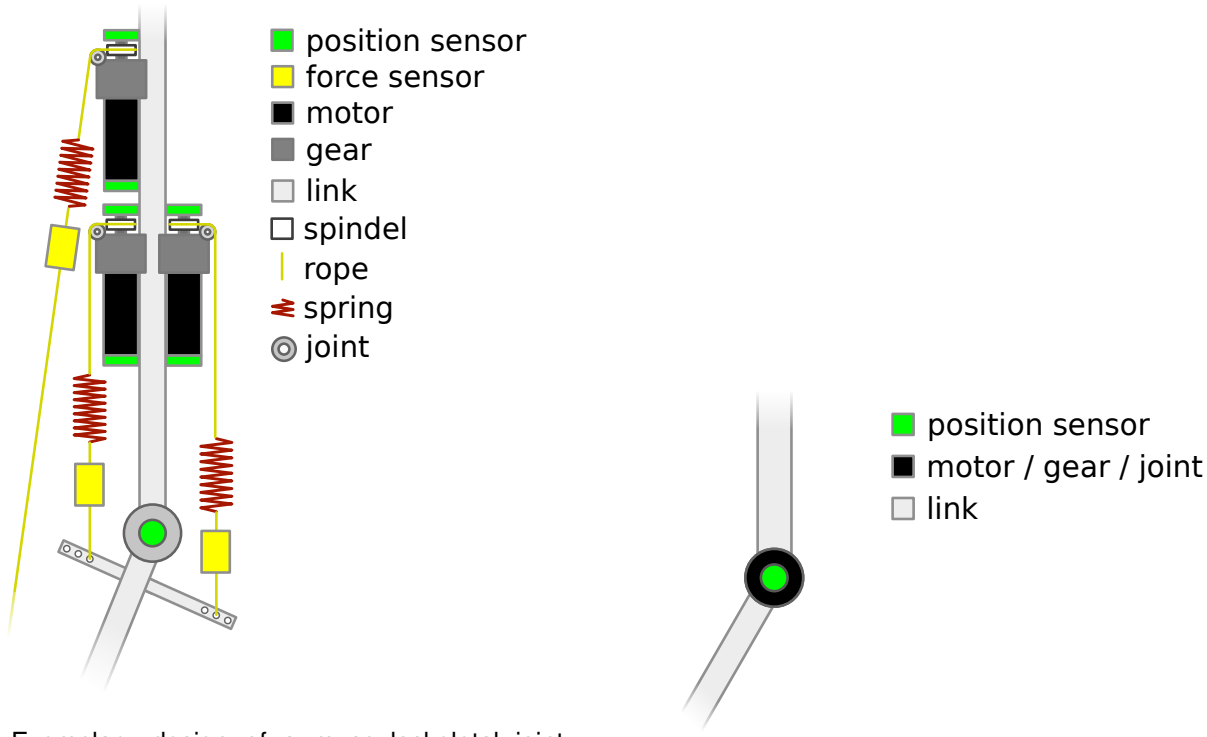
On a prototype platform of a highly elastic musculoskeletal robot a multitude of sensor data are necessary to control the leg joints and the overall posture, but also to collect data for later analysis. The electronic system consists of the components involved in acquiring and transmitting sensor data, calculating the control outputs and moving the actuators based on them. On the lowest level of this system are the sensors measuring values like positions, forces, torque, accelerations and rotations. For a true multi-variable control all these sensor data need to be transmitted to a central processing unit, which runs the control algorithms on them. This processing unit outputs commands for the actuators to perform which then need to be transmitted back to the individual actuators. Control of the actuators should occur with as little delay as possible and at a high rate, so the central processing unit needs to be able to work with low latency under hard real-time constraints. For control approaches based on complex (e.g. multibody system dynamics) models this means being able to do extensive calculations in a short time to hold the high control rate. In case the central processing unit is not directly connected to all sensors and actuators, a low latency, high bandwidth bus system needs to be used to communicate with all of them in real-time. In the early development stages the central processing unit can be external to the robot for ease of development, but has to fit on-board the robot for it to work in untethered operation at later stages.

3.2.1 Sensor Data

Depending on the various objectives involved in the use of the robot (control, motion optimization, data analysis, comparison to human data) different types of sensor or derived data are needed. The required types of sensor and derived data for different purposes can be seen in Table 3.1. More details about the generation and use of each data type are given in the following list.

On a conventional rigid joint actuation with one actuator per joint usually only a single rotational position encoder is needed per joint as the actuator output position equals the joint position (see Figure 3.2(b)). For tendon driven series elastic actuation of a joint the motor and joint angles have to be measured separately as they are only elastically coupled as shown in Figure 3.2(a). Further measurements of the force acting on the joint side of the elastic element are needed to determine each actuators influence on the joint.

Motor angle The motor angle can be measured using a rotary position encoder at the motor axis before the gear and can be used for position control of the motor. As the rotary encoder of the motor is turned more than a whole rotation, it has to be calibrated with respect to the joint angle after each power cycle of the system. The effort needed for this can be reduced using data from the gearhead angle sensor and the joint angle sensor.



(a) Exemplary design of a musculoskeletal joint with flexor, extensor and biarticular series elastic actuators and their sensors.

(b) Stiff actuator for one joint with one rotational position encoder.

Figure 3.2: Exemplary musculoskeletal setup of tendon driven series elastic joint actuation in comparison with conventional rigid joint actuation. In (a) a full musculoskeletal joint setup is shown with extensor, flexor and biarticular actuator, each with two position encoders, a spring and a force sensor. For rigid robots usually one motor per joint acting as extensor and flexor with a single rotary position encoder is sufficient as shown in (b).

Source: own representation

Motor speed The motor speed can be derived from the measured motor angles. For low speeds the quality of the derivation depends on high resolution and low noise of the position encoder. For high speeds the encoder frequency is more important.

Motor torque The motor torque can be measured using torque sensors at each motor or it can be derived by measuring the motor current and applying a forward model of the motor.

Gearhead angle The gearhead angle can be measured using a rotary position encoder at the motor axis after the gear. It can be used for calibration of the motor position after a power cycle of the system. With a tendon driven series elastic actuation based on winding a rope onto a spindle, it is possible for the gearhead to rotate more than a whole rotation. Therefore, the gearhead angle sensor also needs to be calibrated

	Motor control	Joint control	Posture control	Human data comparison	Data analysis
Motor angle	x	x	x		x
Motor speed	x	x	x		x
Motor torque	x	x	x		x
Gearhead angle	x	x	x		x
Joint angle		x	x	x	x
Joint speed		x	x	x	x
Joint torque		x	x	x	x
Spring force				x	x
CoM acceleration			x	x	x
CoM rotation			x	x	x
CoM position			x	x	x
Foot position			x	x	x
GRFs			x	x	x

Table 3.1: Sensor data and derived data required for different control and analysis concepts. (CoM: center of mass, GRFs: ground reaction forces)

with respect to the corresponding joint angle with the help of the joint angle sensor.

Joint angle The joint angle can be measured using rotary position encoders in each joint. These values can be directly used for joint position control. Sensor resolution, frequency and noise are important factors for the quality of the resulting control.

Joint speed The joint speed can be derived from the measured joint angles. For accurate calculation a high resolution, high frequency and low noise encoder should be used.

Joint torque To allow controlling the overall leg like a spring with variable stiffness it is important to be able to control individual joints based on their torque. The joint torque can be measured either using torque sensors in each joint or it can be calculated from all forces acting on the joint.

Spring force The spring forces are an important information when calculating the joint torques based on all acting forces. The force itself can be measured either directly using force sensors at each spring or possibly be derived from the known spring constant and the measured deviation between the joint side and the actuator side position encoders.

CoM acceleration The center of mass (CoM) acceleration can be measured using an accelerometer attached close to the robot CoM, usually located inside the trunk.

CoM rotation The CoM rotation can be measured using a gyroscope attached and is a vital information in any balance control approach. Also, to determine the leg angle during flight phases it is important to know the CoM rotation as a base for forward kinematic calculations.

CoM position The CoM position can be derived from the accelerometer and gyroscope data. Due to the noise and drift of the sensors a filter has to be used and usually supported by a second data source. Here a sensor fusion with the kinematic data during the ground contact phase are possible.

Foot position The foot position can be described relative to the trunk or to the ground. When foot placement is to be used for balance control the foot position relative to the trunk is required. This can be derived from the joint angles using forward kinematics and the trunk position and orientation as base. For landing preparation, obstacle avoidance and locomotion on slopes the position relative to the ground is required.

3.3 Requirements for Monitoring, Configuration and Analysis

The internal sensor data of the robot are not only used for the control of the robot but also for monitoring of the robot during operation as well as for later offline analysis. For the live monitoring of data to be useful during the operation of the robot, soft real-time constraints have to be fulfilled to display the data synchronously with the robot motion. The data for the offline analysis are only recorded during robot operation and have therefore no real-time requirements. But it is important for these data to be recorded with timestamps to allow for synchronization of multiple sensor data sources during analysis, e.g. external force plate or video camera [81].

During the development of control concepts for dynamic motions of a musculoskeletal bipedal robot a large amount of sensor data needs to be monitored to allow for analysis and debugging of the robot's motion. It is of high importance to be able to capture as much internal data of the robot as possible for later analysis to improve control concepts and parameters. While the control process for the robot has to be able to run on-board the robot for autonomous operation, the monitoring of the data can take place on an external system with capabilities for data visualization. As a musculoskeletal robot with its antagonistic and biarticular structures consists of more actuators than a conventional robot with a motor per joint structure, more sensors are necessary to monitor all actuators. But not only the active actuators and joints need to be monitored, also passive structures that influence the robot motions like elastic elements are important. Although it is also possible to generate this data

using a simulation model of the robot, this is not equally valuable. Since there is always a difference between the robot and its model, the data generated directly on the hardware are the most valuable as they are the only exact data.

Besides the monitoring and analysis it is very important for the robot operator to configure the parameters of the controller and calibrate the sensors to facilitate repeatability of experiments. This needs an interface with display and input capabilities for the operator to interact with, which can be run on an external system.

3.4 Requirements on the Software Architecture

A software architecture used for control, configuration and monitoring of a series of prototype robots needs to be able to accommodate all the aforementioned requirements on control, the electronic system and the monitoring, configuration and analysis. Thus, the control algorithms need to be able to run under hard-real time constraints while having low latency high bandwidth access to all sensor data. The monitoring interface, while also needing access to all sensor data, only needs to run under soft real-time constraints, possibly on an external system. It also has to be possible to change the parameters of the controller during runtime through an external configuration interface during the development to quickly adapt the generated motions and calibrate the sensors. Analysis of the recorded data takes place off-line and can therefore be performed separately from the control and monitoring.

Based on these requirements the software architecture for the overall system should allow the separation of the software into multiple components, running in different processes possibly on different systems. The main components of the system are the control component, the monitoring and configuration interface and the data analysis tool.

The control component is tasked with converting the sensor data into physical units and filtering them. Further, it generates derived data and combines data from multiple sensors through sensor fusion. Based on the current system state calculated from this data and possibly model data it calculates a desired state for the robot and generated the necessary control outputs for the actuators.

The monitoring and configuration interface is the main point of interaction with the control software for the robot operator. Thus, it should provide a convenient human computer interface with graphical visualization capabilities for the data monitoring and text based input capabilities for the parameter configuration.

For the offline analysis of the recorded data also graphical visualization capabilities are required. But besides the simple visualization of sensor data values, a programmatic analysis and processing of combined data from multiple sensors or a whole series of experiments should be possible. This also raises the requirements for the visualization of the combined data with higher complexity (e.g. higher dimensionality).

The communication between the control component and the monitoring and configuration processes should be possible over network between multiple systems to allow the separation of the on-board control system from the off-board monitoring and configuration system.

All components in this software architecture need to be easily extensible for use with new control concepts or additional sensor data in new prototype generations.

4 Software and Hardware Design Considerations and Developments

This chapter focuses on the design considerations and the development of the software and hardware of the musculoskeletal robot prototype series BioBiped. The main contributions of this thesis are the new control architecture design to allow full multi-variable control described in Section 4.4 and the mechanical advancement of the robot generations detailed in Section 4.6. Further, a graphical user interface framework was developed as part of this thesis which is now available as open source and is widely used in robotics research as described in Section 4.5.2. All experiments made in the following chapters use the software and control system developed in this thesis, which is described in more detail in the following sections.

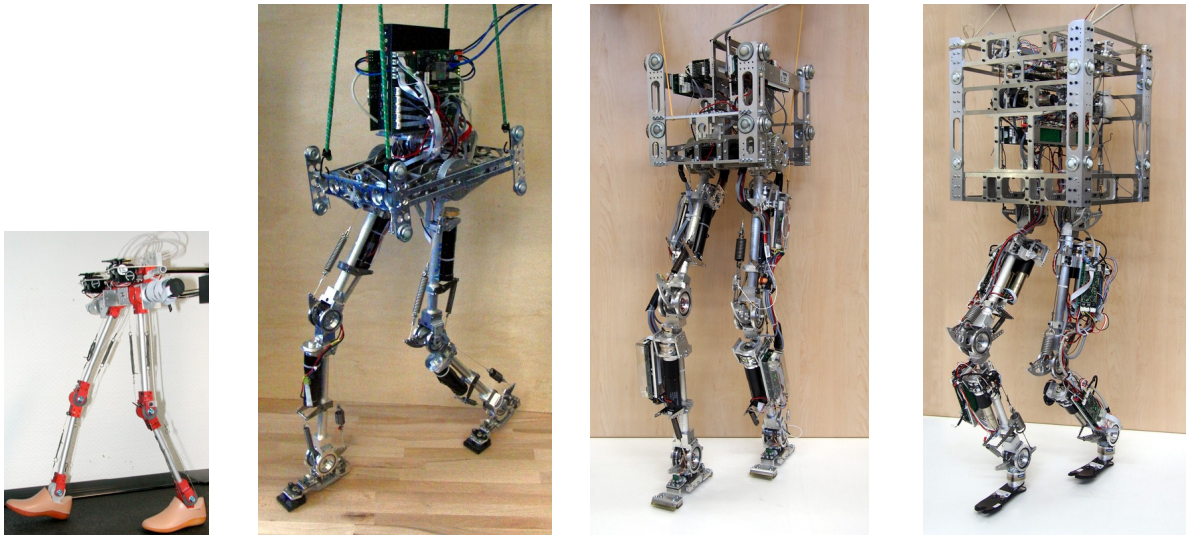
4.1 BioBiped Robot Series

The BioBiped robot series consists of musculoskeletal bipedal robots (shown in Figure 4.1) developed with the goal to perform more human-like locomotion than has been achieved with conventional rigid robots so far. Furthermore, it aims to provide a testbed for experimental evaluation of hypothesis from biomechanics and give insights into the roles of different structures for various leg functionalities required in locomotion. Designed for that purpose it offers the flexibility to change various mechanical configurations like spring stiffnesses, attachment points and the addition or removal of certain structures to compare different hardware setups. Also, it features a vast range of on-board sensors to not only allow for real-time control, but also provide additional data for monitoring and offline analysis as required by Chapter 3.

4.2 Design Concepts

With the focus on human-like locomotion the leg and actuator structures for the BioBiped robots were derived from biomechanical understanding of functional human leg structures. As the three-segmented leg is potentially subject to overextension in its joints, the segment lengths were chosen with a human-like ratio, which helps to avoid this problem according to [73].

To be able to achieve human-like motion performance in running gaits using a robot with comparable power to weight ratio, an elastic actuation system with the potential to store and release energy in its elastic components is necessary. Commonly available technologies for elastic actuation are pneumatic actuators as well as series elastic actuators. While pneumatic actuators are inherently elastic and offer high forces already at slow speeds, they are also non-linear and have a hysteresis, which makes them difficult to control. On the other hand combining an electric motor with a gear and a spring to form a series elastic actuator (SEA, Figure 4.2(a)) allows the use of well-known control concepts for conventional servo motors.



(a) JenaWalker II

Source: [75]

(b) BioBiped1

Source: own photo

(c) BioBiped2

Source: own photo

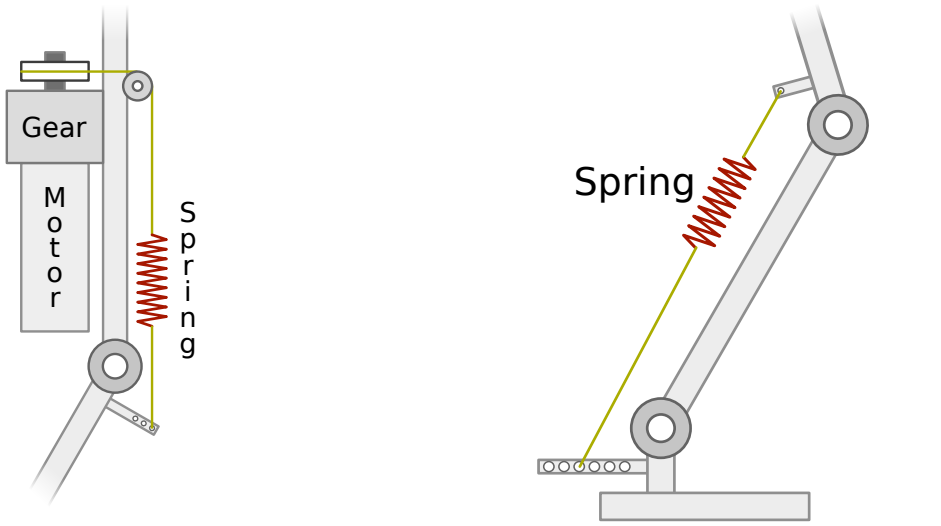
(d) BioBiped3

Source: own photo

Figure 4.1: Different generations of the musculoskeletal BioBiped robot series and its predecessor JenaWalker II.

With the spring in between gear head and the joint, the motor is decoupled from the joint, passively protecting the gears from impacts and allowing the spring to store and release energy independently. In comparison to conventional rigidly actuated joints, the series elastic actuator needs more complex control concepts to achieve similar precision. To be able to reduce the active control effort needed, a musculoskeletal configuration is used which can offer additional benefits through multi-joint coupling. Here multiple joints are coupled using biarticular structures (Figure 4.2(b)) with the potential to passively handle some control tasks on the mechanical level, like power distribution or synchronization between joints. But to benefit from this potential the hardware has to be carefully designed and configured as the multi-joint couplings can also lead to undesired effects making the control of the system even more difficult, e.g. over-determination in the kinematics. So an efficient way to configure the hardware components in combination with the motion control system has to be found as described in Chapter 7. Only then can the biarticular structures be used to facilitate the control of the robot by passively coordinating multiple elastic joints in the desired manner.

In the development of the robot mechanics and its control a multibody system dynamics simulation can help to evaluate and compare different mechanical setups and control strategies efficiently before their implementation on the actual hardware. But since the mechanics of the designed musculoskeletal robot with the artificial muscle-tendon structures and multi-joint couplings are highly complex, modeling its dynamic behavior is only possible to a certain degree. There are still non-negligible differences between the simulation model and the robot as can for example be seen in Figure 7.3 and is further discussed in Section 7.1. Therefore, experiments on the hardware are of high importance for the optimization of the robot's design and its mechanical and control parameter setup as detailed in Chapter 7.



(a) Series elastic actuator as extensor for a joint. (b) Passive elastic biarticular structure.

Figure 4.2: Examples of elastic structures used in musculoskeletal robots.

Source: own representation

4.3 Mechanical Design

The robot design is based on three-segmented leg kinematics similar to human legs and features a combination of mono- and biarticular passive and active structures. Based on insights from the JenaWalker II (see Figure 4.1(a)) a combination of active mono-articular agonists with passive antagonists and passive biarticular structures was chosen for the first prototype. All structures are either passive elastic or active series elastic making the system underactuated, but not completely passive. With this consistent use of elasticity the gears and joints are protected from impacts and the legs are able to store and release energy between rebounds (see 6.1.1). The active extensors are responsible for the power input to the system (see 6.1.2), while the biarticular structures distribute and synchronize the power between the joints (see 7.2). Implementing these mechanical controllers aims to reduce the complexity of the control software needed. Nevertheless, numerous sensors have been mounted to be able to gather information about the systems motions not only for control, but also for analysis.

Three generations of robots (see Figure 4.1) have been designed and built in the BioBiped series based on the results from experimental analysis of their predecessors. The leg links are implemented as bone-like structures with actuators and passive structures attached on the outside, whereas the torso consists of an outer frame with the functional structures on the inside, which allows it to be guided on the outside by an external constraining mechanism. All generations feature the previously described series elastic actuation using electric motors, linear springs and ropes as elastic tendon systems as well as passive elastic biarticular structures. They all have a hip height of 0.7 m in straight standing configuration, but differently sized torsos, all without arms and heads. An overview of the physical dimensions of the robots can be found in Table 4.1 with more details about the changes between the generations in the following section.

	BioBiped1	BioBiped2	BioBiped3
CoM Height [m]	0.57	0.59	0.71
Hip height [m]	0.7	0.7	0.7
Mass [kg]	9.2	11.5	15.9
Degrees of freedom	9	6	6
Number of actuators	9	6	12
Number of sensors	24	27	43

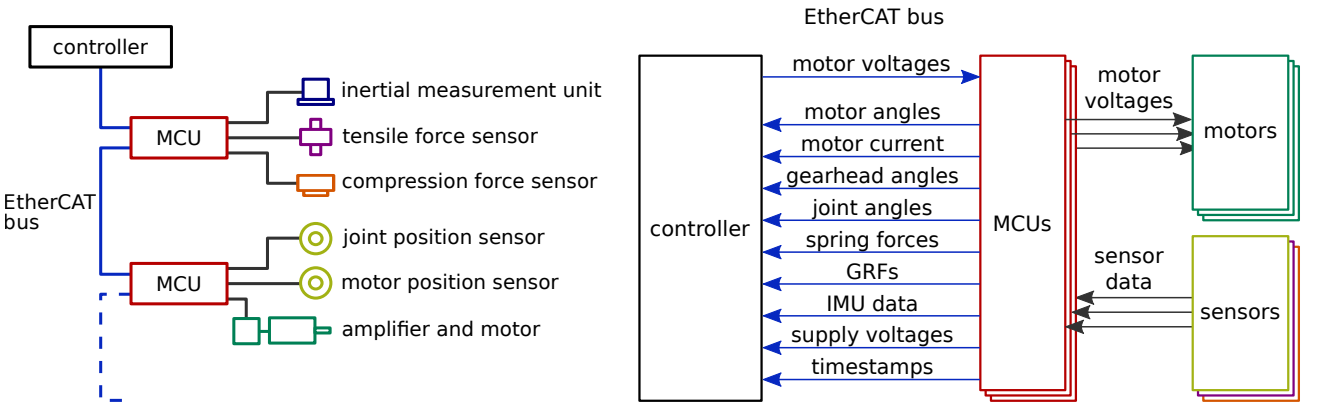
Table 4.1: Overview of physical properties of the three BioBiped robot generations. For details and reasoning behind the changes see Section 4.6.

4.4 Electronic Control Architecture

4.4.1 New Approach for the BioBiped Series

To fulfill the requirements for supporting highly diverse control concepts including a full multi-variable and model based control described in Chapter 3 for the BioBiped robots, an electronic control system is implemented that allows for central real-time control and monitoring of the robot prototypes with numerous additional sensors for analysis of the robot's motions. The system consists of a central control system with enough processing power to allow the implementation of model based real-time control connected via a bus system to multiple MCUs interfacing with the motors and sensors as shown in Figure 4.3(a). While the MCUs are needed for their physical interfaces to the sensors and motors, in this system they only act as relays for reading sensor data and setting motor voltages provided by the central control system in real-time as depicted in Figure 4.3(b). In contrast to the commonly used cascade control design described in Section 2.4, no control task is executed on the MCUs, leaving all power over the control to the central control system. This allows implementing numerous control concepts in software on the central control system, including full multi-variable and model based control. Through the modular design of the MCUs that started with the seconds BioBiped generation, extending the control system with new actuators and sensors is possible without losing the ability to apply centralized multi-variable control concepts. To the author's best knowledge no other musculoskeletal bipedal robot is using a control system with complete low-level real-time access to all sensors and motors from a central control system, while still allowing for the implementation of highly complex control concepts through high processing power and the extension of the electronics with additional modules.

As central control system either an embedded computer mounted on the robot can be used to allow autonomous operation or an external computer providing even more processing power can be utilized while tethered during development. To be able to access and control all low-level sensor and motor data as required for full multi-variable control in real-time a high bandwidth, low latency bus system is used to connect the central control system and



(a) Control system bus

(b) Control system data flow

Figure 4.3: Control system bus and data flow diagrams. Note that contrary to conventional implementations and despite the distributed design of the electronics in (a), the control itself is not distributed, but handled at full control rate by the central controller based on the low-level data from all sensors as shown in (b). The micro controller units (MCUs) act only as data relays between the sensors and motor amplifiers (AMPs) and the controller to allow the implementation of true multi-variable and models based control approaches. (IMU: inertial measurement unit, GRFs: ground reaction forces)

Source: own representation

the micro controller units. The EtherCAT bus system used here is based on the 100BASE-TX Fast Ethernet standard and allows for an effective use of more than 90% of the full-duplex 100 Mb/s bandwidth [10]. Together with a cycle frequency of up to 30 kHz this performs far better than other possible choices like e.g. the widely used CAN bus. Furthermore, to connect to this bus system only a standardized Ethernet port is needed, offering a wide range of choices for the control computer.

4.4.2 Sensors

Based on the requirements for the sensor data specified in Section 3.2.1 and the results from various experiments described in Section 4.6 the sensors for the BioBiped robots were chosen for the different generations.

Rotary Position Encoders

To sense the joint and gearhead output angles, rotary position encoders are needed. In order to reduce the effort necessary for the calibration process of the robot after a power cycle, absolute position encoders are chosen over incremental ones. To prevent degradation of the sensor through wear, contactless sensors are preferred over mechanical encoders. Among the optical, magnetic and capacitive absolute encoders the choices are manifold. Guided by resolution, physical size and available electronic interfaces magnetic absolute encoders based on

the Hall Effect were chosen. Over the three generations of BioBiped robots the resolution of the used sensors was improved from $0.35\,\mathrm{deg}$ to $0.09\,\mathrm{deg}$ to allow for better derivation of angular velocities. Further, the sensor interface was changed from analog measurement to digital readouts through the Serial Peripheral Interface (SPI) bus to avoid sensor noise induced in the wiring between sensor and MCU.

Motor Position Encoders

For the control of the motor positions an integrated incremental rotary encoder on the backside of the motors is used. The communication with the MCU is handled digitally using a Transistor–transistor logic (TTL) interface.

Inertial Measurement Unit

To sense the inertial motion of the robot's trunk including acceleration and rotation an inertial measurement unit (IMU) is mounted on the trunk as close to the center of gravity as possible. A IMU including three axis acceleration and three axis rotation measurement is used to allow for full tracking of the trunk's posture for balance and posture control. The IMU module chosen combines these into a single casing and interfaces with the MCU digitally over the SPI bus which prevents noise induced on the wiring.

Ground Contact Forces

To detect ground contact and also measure the forces acting between the feet and the ground force sensing is implemented in the feet of the BioBiped robots. In the first two generations a custom design based on slightly bending steel plates and Hall Effect sensors measuring this deflection are used. Due to mechanical problems described in Section 4.6 and a difficult calibration process they were exchanged for dedicated six-axis force torque sensor.

Spring Forces

Direct measurement of the forces acting at the springs of the series elastic actuators was implemented starting with the BioBiped2 robot. The tensile force sensors can be mounted directly in the rope connecting the spring and the joint. Due to the size constraints this involves the data is read out as a differential analog signal with the conversion to digital data on the MCU.

4.4.3 Actuators

The series elastic actuators (SEAs) used in the BioBiped robot series are a combination of electric motor, gear, spindle, rope and spring as can be seen in Figure 4.2(a). Dimensioning

of motor and gear are based on simulation studies performed in [62]. Through exchangeable spindles with different diameters and changeable lever arm lengths on the joint side the effective gear ratio can still be adjusted on the robots. Further, the springs are also interchangeable and are subject to the optimization process of the hardware setup detailed in Section 7.3.

4.5 Software Components

To fulfill the requirements set in Section 3.4 and also support an efficient development cycle the software used to control, monitor and analyze the robot's motions is split into components. These components can run on a single computer system for ease of development, but can also be distributed over multiple computers connected via a TCP/IP network. This allows the robot to be controlled via an on-board computer which is monitored and configured via network by the operator, making untethered robot operation possible. The software components developed in this work combine various existing software technologies with newly implemented solutions to fulfill the requirements described in Chapter 3.

4.5.1 Used Existing Technologies

Communication between the software components of the control and monitoring system is handled through the Robot Operating System (ROS) middleware [58, 65]. This allows for the modularization of the system into multiple nodes, which can be run on a single or on multiple systems connected via network. This fulfills the requirements from Section 3.4 to adapt the system depending on the current use case e.g. ease of development, more processing power during experiments or untethered operation of the robot. The ROS middleware was chosen over competing solutions like the Orocos toolchain for its large user base which provides a better chance of long-term support.

The controller component is written in C++ [27, 78, 85] using Orocos Real-Time Toolkit (Orocos RTT, [49]) for its abstraction of the actual real-time implementation used in the kernel such as RTAI [67], Xenomai [86] or preempt-RT [36]. This allows fulfilling the requirements from Section 3.1 in terms of real-time control and allows the choice of real-time implementation to be changed later on, should problems with one implementation occur. In this case the use of a preempt-RT Kernel [36] on an Ubuntu Linux system allows the controller to run in real-time mode in user space, without any need for application code in kernel space, making the system more reliable during development.

The controller component communicates with the robot control system electronics via the EtherCAT bus with the help of the IgH EtherCAT Master [25] for Linux running its network driver module in kernel space, while offering the transfered data to the control software via a user space API. The IgH EtherCAT Master offers generic network kernel drivers to allow operation on all Linux supported Ethernet interfaces as well as specific kernel drivers for individual EtherCAT chips for better latency performance.

For user interaction and data monitoring as specified in Section 3.3 a graphical user interface (GUI) is implemented in Python [56, 82] using platform independent graphical components from the Qt toolkit [57] via PyQt [55]. Using a scripting language like Python here allows for rapid application development [79] including fast adaptation to requirements of the new robot prototype generations thought fast testing cycles, while its reduced runtime performance is still good enough for user interfaces. Python is also the second officially supported programming language for the ROS middleware, making it the ideal candidate for this purpose.

To visualize data during offline analysis as required in Section 3.3 a software is implemented based on the Python matplotlib library [24, 42], which allows for flexible batch processing of many data sets and produces high quality graphs for publications.

4.5.2 Own Software Developments Released as Open Source

At the beginning of this work the ROS ecosystem did not include a consolidated and integrated GUI framework for configuring, controlling and monitoring robots, but only a collection of various graphical tools for specific tasks. As the author had experienced the benefits of integrated GUIs before in other projects [51, 81], a new integrated GUI framework for ROS was developed during this thesis to provide the features needed in this project. This work was the starting point for the rqt project [80] which is published as open source and now is the standard GUI in the ROS ecosystem [40] used and extended by many robot developers worldwide [2, 6, 9, 31, 66]. It offers the basis for customizable GUIs by allowing to combine widgets from various plugins into a main window offering management of different window layouts for different tasks. The plugins can be implemented using either Python to allow for rapid development or C++ for better runtime efficiency depending on the requirements for each plugin. Through the use of the Qt graphical toolkit is it possible to run plugins of both variants simultaneously in one integrated GUI window.

4.5.3 Hardware Abstraction Layer

As described in Chapter 3 the requirements in this project for the abstraction of the robot hardware are quite different from most other robotic projects, as it is defined in the project goals, that the robot prototype generations will differ significantly from each other. Not only will the types of sensors and actuators change, as it is common between robot generations, but also the number of actuators, sensors and degrees of freedom will increase as the robots are planned to perform more complex motions. To accommodate these requirements without having to rewrite the control software for each generation a hardware abstraction layer is very important. Furthermore, it enables the use of the same control software also to control the multibody system dynamics simulation model [59, 61] implemented in MATLAB [41] using Simulink [77], allowing for an easy transfer of simulation results to the robots. To allow the use of the controller component with different generations of the robot prototype this abstraction layer performs the conversion between sensor and motor data and physical units. It

also maps joint and sensor names to devices addresses on the EtherCAT bus (or MATLAB API) and memory addresses in the control packets. This makes the controller component agnostic to a high degree about the system specifics of the robot prototype it is controlling. While it still needs to know about the existence of motors and sensors to use them, their actual type can be changed transparently.

4.5.4 Control Component

This is the central component for the control of the robot, which needs to be running under real-time constraints to allow for uninterrupted control output generation. Further it needs to offer an interface to the GUI component to allow the operator monitoring and controlling the robot using the GUI. Therefore, it has been implemented as an Orocos RTT component with a ROS node as the interface towards the GUI, allowing communication over a network. The control rate this component is run at can be manually configured depending on the requirements of the control approach used. Limiting factors to the control rate are either the EtherCAT bus bandwidth at approximately 30 kHz or the processing power of the control computer depending on the computational complexity of the control algorithm.

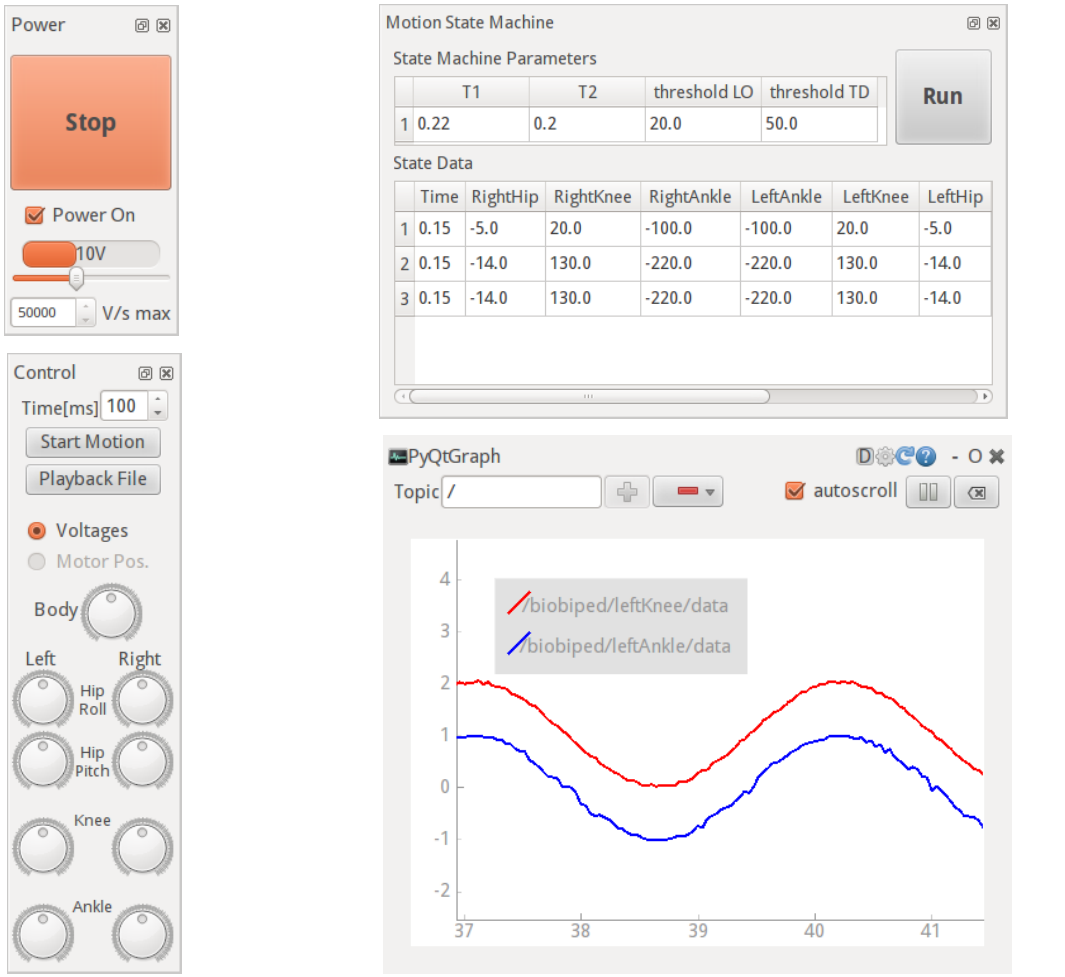
To be able to evaluate different control concepts, the control component offers a programming interface to allow for different controller implementations to be used as control modules. During robot operation the operator has the possibility to change the currently running control module, as well as its parameters, allowing for efficient test cycles.

4.5.5 Monitoring and Configuration Interface

The main interface for the robot operator to interact with the robot's control parameters is a graphical user interface (GUI) newly developed in this work and released as open source. As described earlier in this section it is based on Python and Qt for rapid development and integrated into the ROS middleware for data exchange over a network. Its development also lead to the rqt project described above which offers an integrated GUI allowing combining multiple widgets with different functionalities into a common interface. The widgets developed for the operator to configure and monitor all parameters of the control component of the Bio-Biped robots can be seen in Figure 4.4. Each parameter can be monitored graphically using the plot widget during robot operation to allow for a fast identification of problems and easy tuning of parameters. Also, the parameters of the state machine described in Section 5.1.2 as well as the conditions for the transitions can be configured.

4.5.6 Data Analysis

A second graphical tool has been developed to help with the offline analysis of the data recorded during experiments. Its focus is the graphical visualization of sensor and derived data in two dimensional plots, but it also offers additional visualization like a forward kine-



	RightHip	RightKnee	RightAnkle	LeftAnkle	LeftKnee	**LeftHip**
Timestamp	23890.0146	132369.1570	23888.7382	0.0010	0.0010	
JointAngle	-14.5020	0.0000	-59.2822	-64.3799	39.7851	-14.9122
MotorAngle	-11.7676	130.6482	-200.0259	-200.7953	130.3553	-11.6603
MotorCurrent	0.0000	0.0010	0.0020	0.0000	0.0020	0.0010
HallSensor	5.3400	1.2700	0.1900	0.2200	-6.6500	2.2500
ForceSensor	-0.1120	-1.7070	6.3224	0.0480		
Gyroscope	-14.0000	-24.0000	18.0000			
Accelerometer	29.0000	5.0000	1010.0000			
SpringLength	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SystemVoltage	24.1000	24.1000	24.0000	24.1000	23.9000	24.1000
ControllerGoalAngle	-12.0000	130.0000	-200.0000	-200.0000	130.0000	-12.0000
MotorVoltage	0.4647	1.2964	-0.0517	-1.5906	0.7106	0.6794
GoalAngle	-5.0000	40.0000	-100.0000	-100.0000	40.0000	-5.0000
ControllerPGain	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
ControllerIGain	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
ControllerDGain	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
ControllerMode	4	4	4	4	4	4

Figure 4.4: Some widgets of the graphical user interface used to monitor and (re-)configure the robot during operation.

Source: own representation

matic view as can be seen in Figure 4.5. Further, different view arrangement can be saved as presets to batch process multiple data files into a number of plots suitable for publication.

Also, it functions as testbed for data filtering and derivation algorithms that are to be implemented directly into the controller later on like a Kalman filter for pose estimation of the trunk based on the IMU data. Quick development in Python with the help of the NumPy mathematical library allows for prototyping of the algorithms and visualizing their output based on recorded data before implementing them in C++ for use in the real-time controller.

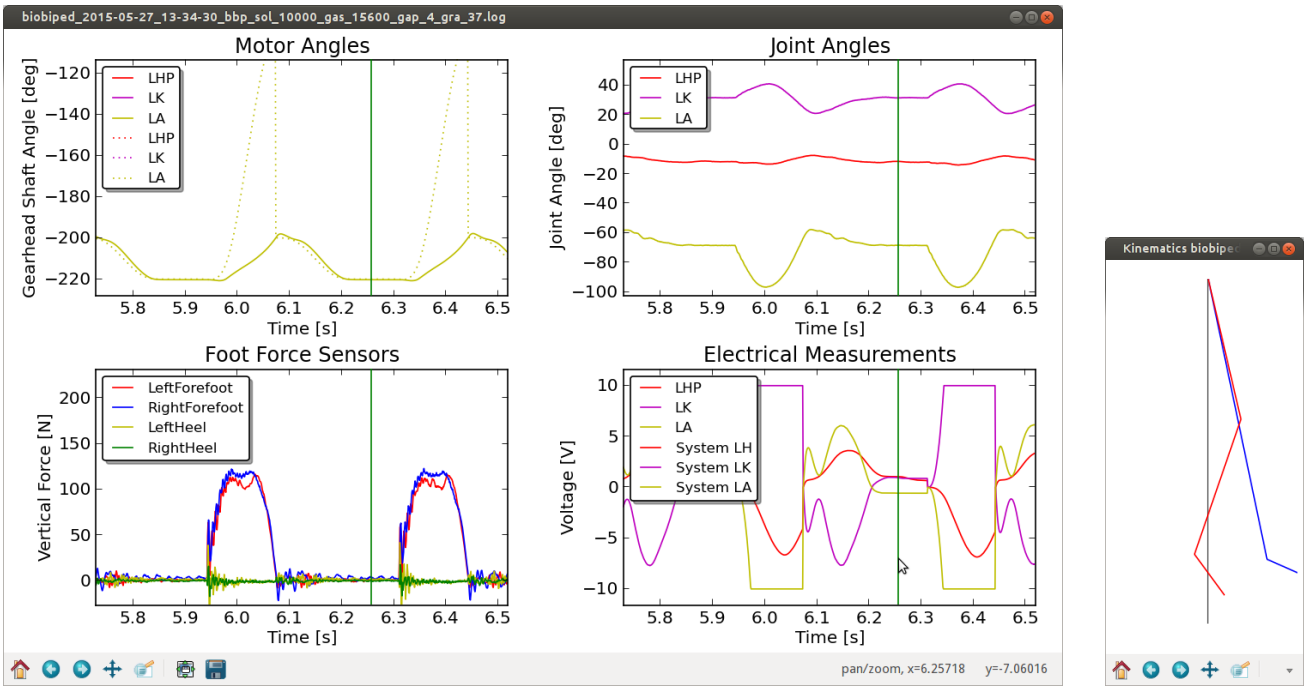


Figure 4.5: Graphical user interface used for the offline analysis of sensor and derived data.

Source: own representation

4.6 Functional Evolution of the BioBiped Generations

The first robot design in this series called BioBiped1 (see Figure 4.1(b), [60]) was developed based on the highly underactuated JenaWalker II (see Figure 4.1(a), [75]), which performed walking and jogging motions on a treadmill while being constrained and supported externally. The JenaWalker II has only one actuator per leg which is coupled with the hip, knee and ankle pitch joints through ropes and springs. Careful tuning of these rope lengths and spring stiffnesses was needed for it to successfully perform different gaits [75].

To gain more control over the movements of the individual joints, BioBiped1 has one actuator for each joint. It has rotational joints in the ankle, knee, hip and trunk all rotating about the pitch axis and in the hip additionally around the roll axis. In case of the hip joints the motor is coupled with the joint in both directions with elastic elements between motor and joint forming a bidirectional SEA. At the knee and ankle joints on the other hand the motor is

connected with the joint only in one direction allowing it to extend the joint, while flexing it is achieved by a passive spring.

The changes made to the different generations based on the results of experiments made with their predecessors are described in the following sections.

4.6.1 Mass and Inertia Distribution

BioBiped1 and 2 have rather small and lightweight torsos compared to their total height and mass, making their legs relatively heavy and giving them a low center of mass (CoM).

Problem in BioBiped2

This poses problems when trying to implement foot placement in the flight phase due to inconvenient inertia ratios between leg and torso. Further, having the CoM below the hip joint is not human-like and disallows the use human-like stability concepts like the virtual pivot point (VPP, [43]).

Solution in BioBiped3

By increasing the dimensions and mass of the torso in BioBiped3 (see Figure 4.1) a larger inertia is created to allow for more stability in the torso during leg placement in the flight phase. Further this moves the center of mass of the robot above the hip axis, which is a prerequisite for some stability concepts like the VPP. The mass is not increased through dead weight, but rather by adding motors to the torso which are needed for more leg position control during flight phase described in the next section. To keep the overall weight low despite doubling the number of actuators, all motors were changed to brushless motors, which weight only 168 g (compared to 260 g) while offering more power output.

4.6.2 Foot Design

Problem in BioBiped2

Vertical ground reaction forces measured for BioBiped2 hopping on a force plate showed a peak impact force just before the loading of the leg (Section 6.2). This impact force is in relation to body mass much higher than for humans performing similar motions and was attributed to the relative mass and stiffness of the BioBiped2's feet when compared to human feet. A foot of BioBiped2 accounts for 15.2 % of one leg's mass (3037 g), while according to [8] in humans the foot provides only 6.3 % of the leg mass. Further, the BioBiped2 foot is stiff, making the extensor spring of the ankle joint the first structure in the kinematic chain to damp the impact. In contrast, the human foot itself has internal damping properties already below the ankle joint.

Solution in BioBiped3

For BioBiped3 the foot design has been completely revised to now consists of a standard prosthetic foot attached to a force-torque sensor just below the ankle joint. The prosthetic

foot is made of two Carbone fiber leaf springs (one for the forefoot and one for the heel) which allow damping of the impact already in the foot. Also, the mass below the ankle of now is only 11.3 % of the total leg mass bringing it closer to a human mass ratio compared to the previously used aluminum design. The ability, to additionally put a damping training shoe around the prosthesis, should help to reduce the peak impact forces on touch down even further.

4.6.3 Actuated Structures

The design of the BioBiped robots started out from its highly underactuated predecessor JenaWalker II and added only active structures to the knee and ankle extensors for power generation. With the advancement from hopping to jogging motions shortcomings of this still underactuated design were found. Furthermore, the interest in evaluating biomechanical hypotheses about the active functions of biarticular muscles was limited by the low available number of actuators. Therefore, additional actuators were added as detailed in the following problem and solution descriptions.

Knee Flexor

Problem in BioBiped2

In dynamic hopping motions with single support phases performed with BioBiped2 (see 6.2) the swing leg has to be held in a retracted position to not touch the ground. But during the impact of the stance leg the combined forces of gravity and the impact dynamics are pulling against the passive knee and ankle flexors thereby extending the leg. Using only passive flexor structures to hold against these forces showed to be difficult, especially in the knee with the higher mass of shank and foot to hold up. This resulted in unwanted ground contact during alternate hopping and more importantly jogging motions, where it can lead to stumbling of the robot.

Solution in BioBiped3

Implementing a stiffer spring as flexor could reduce this problem, but would also increase the force needed by the extensor pulling against it. To allow for better holding of the leg posture without increasing the flexor's stiffness to unfeasible values, further actuators were added in BioBiped3. In total six actuators were added (four of them in the torso) to allow for more configuration possibilities of which structures to actively actuate for different motions. Further, the additional motors in the trunk increase the CoM height and the inertia of the torso which, as described already, is required for postural stability control and foot placement. The new actuators can be used to actively actuate the previously passive knee flexor to prevent unwanted ground contact of the swing leg foot.

Biarticular Structures

Problem in BioBiped2

The biarticular structures in BioBiped2 are only implemented as passive elastic elements spanning two joints, which allows tuning them to a specific length and stiffness to support power distribution between joints. But according to biomechanical hypotheses ([45]) humans also use their biarticular muscles to actively adjust leg posture during standing still for postural stability, which cannot be achieved with passive structures.

Solution in BioBiped3

To allow testing this biomechanical hypothesis on a mechanical system, the new actuators in BioBiped3 can be connected to the biarticular structures in thigh and shank to change their length dynamically.

Hip Actuation

Problem in BioBiped2

The actuation of the hip joint in BioBiped2 is implemented with one motor as a full SEA which can turn the joint both ways. Hereby the series elasticity is achieved through compression springs between two discs, one actuated by the motor and the other one connected to the joint. This closed elastic system offers elasticity while taking up only little space in the torso, but is mechanically limited to 10 degrees maximal joint deflection. Furthermore, the springs cannot as easily be exchanged as is possible in the other actuators used in this robot, since they are completely enclosed in the mechanism and have to be of a specific length and diameter to fit in.

Solution in BioBiped3

As BioBiped3 has a larger torso, the hip joint actuator was changed to also use a combination of rope and springs to elastically actuate the hip joint in both directions. The design is similar to the one used in the other joints and the springs can be more easily exchanged with a wide variety of extension spring of different stiffnesses and dimension. Here the maximal joint deviation is only limited by the spring properties and can be up to 90 degrees, depending on the spring and lever arm length.

4.6.4 Transmission Ratios

In the actuation design used for knee and ankle joint (see Figure 4.2(a)) the overall transmission ratio between motor and joint depends on the motor gear ratio, the spindle diameter, the lever arm length at the joint and the current joint position.

Problem in BioBiped1

In the alternate hopping experiment performed with BioBiped1 (see Section 6.1.4) it was found that the retraction speed of the swing leg was limited by the rope speed.

Solution in BioBiped2

To increase the rope speed in BioBiped2 the motor gear ratio was reduced from 66:1 to 51:1 and the diameter of the spindle was made configurable between 24 mm and 36 mm. While this increases the maximum rope speed by up to factor 1.94, it also reduces the maximal force which the motor can apply to the rope. As some motions might need a higher torque rather than higher speed, this ratio was made even more easily adjustable by increasing the range of the usable lever arm length. This allows adjusting the overall transmission ratio within a wider range than before allowing for high torque or high speed setups, depending on the desired motion.

4.7 Evolution of the Robustness and Maturity of the BioBiped Generations

The mechanical robustness of the robots was evaluated using different experiments detailed in Chapter 6. Based on the results from the these experiments the following list of issues and solutions was used to improve the mechanical robustness and maturity of the system over multiple robot generations.

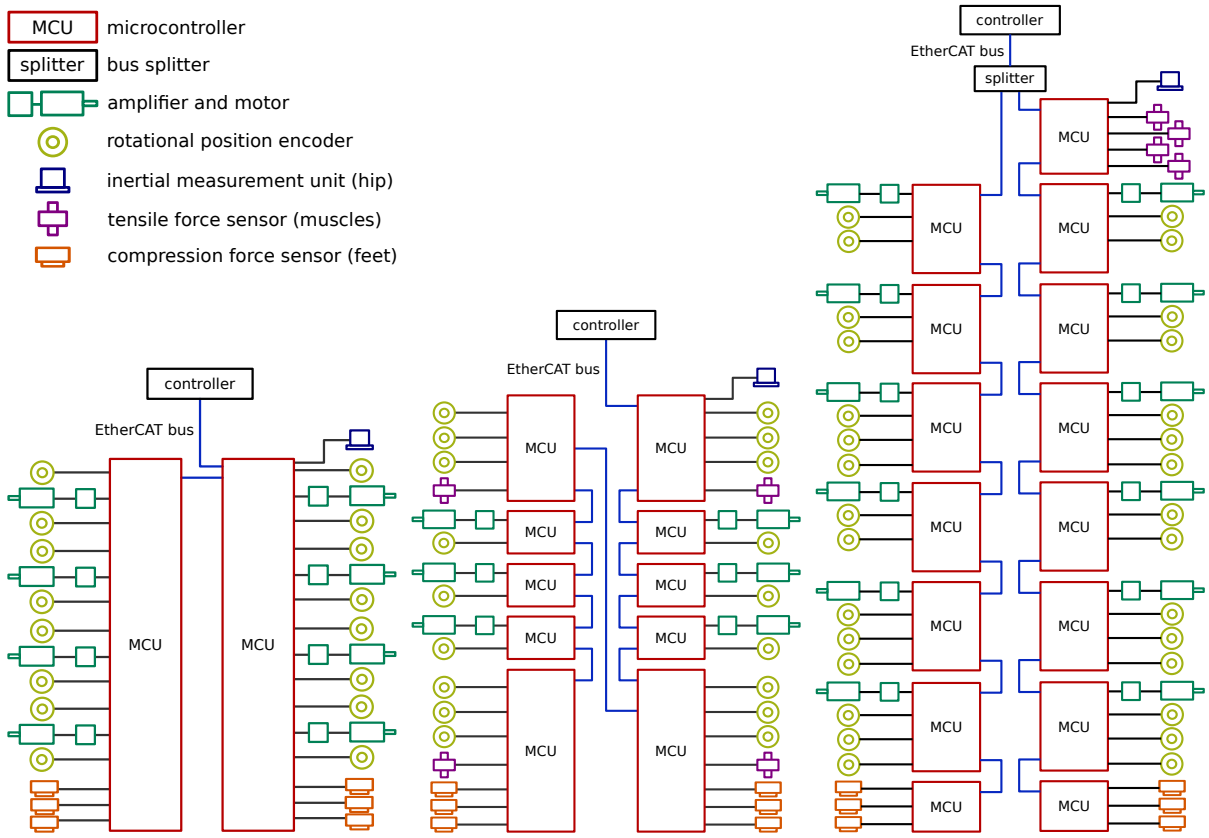
4.7.1 Rope Guiding Pulleys

Problem in BioBiped1

Even though BioBiped1 performed well in the first experiments, some mechanical issues were found over a few months of experiments. These were addressed first by an updated design for the BioBiped1 knee extensor actuation and later more thoroughly in the BioBiped2 (see Figure 4.1(c)) design. The synthetic ropes leading from the motor spindles to the knee and ankle joints were guided around two corners by plastic pulleys with a diameter of 4 mm mounted on plain bearings. Through the high speed of the rope movements and the high tension of the rope when moving around the corners, the pulleys were not always able to turn and the rope slipped over the pulleys. This led to heating of both the pulleys and the rope which caused the synthetic ropes to break and even sometimes the pulleys.

Solution in update to BioBiped1

To avoid the breaking of the plastic pulleys, they were replaced by metal ones. This also reduced the amount of breaking ropes, but especially at the knee this problem persisted. In an update to the knee extensor actuation in BioBiped1 the two guiding pulleys were replaced by a single pulley with a diameter of 12 mm mounted on ball bearings which allows the pulley to turn with less friction even under high load and therefore avoids slipping of the rope on the pulley. This change reduced the amount of broken ropes and was consequently also made to all other joints in the next robot generations.



(a) BioBiped1 (b) BioBiped2 (c) BioBiped3

Figure 4.6: Control system bus of the three BioBiped generations. See Figure 4.3 for information about the data flow.

Source: own representation

Problem in BioBiped2

Through the increased load per leg in the alternate hopping and jogging experiments performed on BioBiped2, the bracket holding the rope guiding pulleys for the knee and ankle extensors started to deform.

Solution in BioBiped3

A redesign of the bracket for BioBiped3 (see Figure 4.1(d)) was done based on analysis of the internal stress of the bracket under relevant load cases using finite element method (FEM).

4.7.2 Roll Joints

Problem in BioBiped1

The roll joints at the hip of BioBiped1 showed to not be mechanically robust enough and developed backlash during the hopping experiments causing the feet to unintentionally move sideways.

Solution in BioBiped2

As BioBiped2 is meant to be externally constrained to translation in two dimensions at the torso, those roll joints are unnecessary for this version. They were removed in BioBiped2 to gain more mechanical robustness, but with the option to add them in later versions when needed.

4.7.3 Joint Bearings

Problem in BioBiped1

The plain bearings in the ankle and knee joints of BioBiped1 degraded after a few months of experiments. Their friction increased over time and also became dependent on the current joint angle. This made modeling and control of the BioBiped1 robot unnecessarily difficult, as the parameters changed over time.

Solution in BioBiped2

By replacing the plain bearings with ball bearings in BioBiped2 the friction was reduced and also made more constant over time. While friction in joints is generally seen as a negative effect when it comes to energy efficiency, the damping it gives at the joint level can also help in making the system more stable to control. Therefore, it was discussed if adding intentional damping to BioBiped2 would be necessary, but the experiments made so far without additional damping have shown the robot to work just as well as BioBiped1 in terms of system control stability.

4.7.4 External Constraining Mechanism

Problem in BioBiped1

In BioBiped1 the torso was built around a central supporting structure and then cased in a frame helping to guide the torso's motion along an external constraining mechanism. Due to high forces between torso and constraining mechanism especially in failed experiments this additional frame was deformed and had to be repaired often.

Solution in BioBiped2

The removal of the hip roll joints in BioBiped2 facilitated a design of the torso based on an external supporting structure which directly worked as a guiding mechanism, making the connection to the constraining mechanism more robust.

4.7.5 Mechanical Robustness of the Foot

Problem in BioBiped2

The foot used for both BioBiped1 and 2 is made from three main aluminum parts in forefoot, mid-foot and heel connected through spring steel plates. This allowed for a simple force

measurement using a hall-effect sensor to detect the elastic deformation of these spring steel plates. But with the higher load of single leg support during alternate hopping and jogging and an increased weight of BioBiped2, the spring steel plates in the fore foot were not strong enough to withstand deformation over a longer series of single legged hops.

Solution in BioBiped3

So in BioBiped3 the feet were replaced by elastic prosthetic feet designed for children of up to 45 kg and the force measurement was moved to a force torque sensor located between this prosthesis and the ankle joint.

4.7.6 Repeatable Calibration

Problem in BioBiped1

The high-resolution motor encoders included in the motors only report the position changes incrementally. Therefore, a power-cycle of the robot resets the values and the motor positions have to be re-calibrated in BioBiped1.

Solution in BioBiped2

To avoid this time consuming process in BioBiped2 absolute rotational encoders were mounted on the motor spindles winding up the rope allowing for repeatable calibration between power-cycles.

4.7.7 Electronic Control System Design

Problem in BioBiped1

The electronic control system in BioBiped1 consists of two large micro controller units (MCUs) each with outputs for five actuators and the same number of rotary position encoder inputs. Through the physical size of these MCUs the robot was limited to two of them and therefore limited in terms of the number of actuators and sensors that can be connected. As described earlier in this section, the number of actuators and sensors was to be increased for various reasons in the next versions of the robot. Subsequently, the design of electronic control system had to be changed to accommodate these updates.

Solution in BioBiped2

Starting from BioBiped2 a more modular approach was used with one MCU per actuator and a few additional ones for extra sensors as shown in Figure 4.6. But the role of the MCUs still remained that of a simple relay, leaving the central controller in charge of control of all actuators based on all low-level sensor data as shown in Figure 4.3. With this modular system the increase of actuators and sensors in BioBiped3 was possible to allow for the active actuation of more elastic structures.

5 Control Concepts for Musculoskeletal Bipedal Robots

The control of a bipedal robot can be divided into a lower level for joint control and a higher level for gait generation and control, which are described in the next sections. Depending on the desired control strategies it can be possible to separate these levels of control into a hierarchy of controllers (e.g. gait generation and joint level control) or it may be necessary to combine them into a single controller for e.g. model based feed-forward control as described in Section 5.3.

For the BioBiped robot series the control system is implemented based on the requirements formulated in Section 3.1. To allow for true multi-variable control the control system in the BioBiped robot series uses a central processing unit with low-level data access as described in Chapter 4. This allows performing all levels of control in a single process and therefore enables the control software to be implemented according to any possible control strategy.

5.1 Gait Level Control

Gait level control is responsible for the generation of motions for the whole robot. Desired motions can for example be described as center of mass trajectory or ground reaction force trajectories.

5.1.1 Requirements and Challenges for Gait Level Control of Musculoskeletal Robots

To achieve dynamic motions goals desired in bipedal musculoskeletal robots it is important to be able to exploit the elastic properties of the legs. In order to do so, the gait level control has to produce not only desired joint trajectories but additional parameters like desired joint stiffness and pretension. This has then to be translated by the joint level control into inputs for the involved actuators, which can be quite complex as described in the Section 5.2. Otherwise, important properties like the rest length of active elastic structures or their pretension cannot be exploited to improve the performance of the robot.

The contribution of individual joints to the overall leg motion is not yet clear. It is possible to use the human joint motions as templates to build upon when generating desired motions.

5.1.2 Approaches for Gait Level Control

Optimal Control

Optimal control is used to generate system inputs that produce a motion of the robot which perfectly matches the desired trajectory. To achieve this an inverse dynamics model of the

robot is used to optimize the system inputs. Due to the complexity of musculoskeletal robots with the interdependencies of multiple joints through biarticular structures, the series elastic actuators and the interactions with the environment this inverse dynamics model would be highly complex. An analytically derived forward dynamics model of a simplified version of the BioBiped2 robot has been created in [12], which is already of high complexity. Creating an even more complex inverse dynamics model of the full robot is therefore not a feasible approach.

Parameterized Trajectory

In a parameterized trajectory generation a periodic signal is produced that is used as target trajectory. Typically, the function used is a sine wave with additional parameters to adjust the resulting amplitude A, frequency f and phase shift φ:

$$y(t) = A\sin(2\pi ft + \varphi) \tag{5.1}$$

This can be used to generate a trajectory for a periodic motions like jogging or hopping. However, it does not account for changes in the trajectory frequency due to disturbances e.g. in the ground contact and is therefore not an ideal candidate for application to musculoskeletal robots.

State Machines

State machines have been used in various areas of robot control, e.g. for trajectory generation in elastic bipedal robots.

Raibert used a centralized control approach with a single state machine controlling all legs in various gait patterns for his different legged telescopic hoppers [63]. The states describe different phases of the gait (flight, landing, compression, thrust, unloading) and contain actions for the actuators to perform to generate the desired leg motion. Transitions are triggered by events detected from the sensors of the robots like leg shortening or maximum leg extension.

On the series elastic planar walking robot Spring Flamingo a state machine is used (see [54]) to describe the high level phases of a walking gait like double, left and right support. Here the state machine is not used to directly generate the legs' trajectories, but rather to change between different control parameters used in the different phases of the gait. The transitions are based on the detection of changes in the legs' ground contacts.

A different use of state machines for elastic bipedal robot MABEL is show in [50]. It switches between states based on sensor detected events describing unexpected disturbances such as stepping up, down or tripping. The states here are not directly involved in the trajectory generation for the walking gait, but rather initiate different controllers used to recover from the detected disturbance.

On the BioBiped robot series a configurable state machine system is used to generate desired trajectories with states corresponding to different phases of the current gait and transitions based on the occurrence of events. But unlike Raibert's state machines described in 5.1.2 the states here do not contain actions, but rather a data set with reference values. The state machine does not know about the semantics of these reference values, as this is handled in the lower level control system, allowing the same state machine to generate desired trajectories for all measurable data, ranging from motor or joint positions (which is not equivalent in a series elastic system) to motor or joint torques. Since direct switching of reference values can lead to discontinuities in the control signal p_{goal}, a smoothing function $p(\tau)$ was implemented. This function transitions the reference value from the currently measured sensor p_{start} value toward its desired value p_{end} over time T with the acceleration limited by a_{limit}:

$$a_{limit} = 8 \cdot \frac{p_{end} - p_{start}}{T} \tag{5.2}$$

$$p(\tau) = \begin{cases} a_{limit} \cdot \frac{2\tau^2}{3} & 0 \leq \tau < 0.25 \\ a_{limit} \cdot (-\frac{2\tau^2}{3} + \tau^2 - \frac{\tau}{4} + \frac{1}{48}) & 0.25 \leq \tau < 0.75 \\ a_{limit} \cdot (\frac{2\tau^2}{3} - 2\tau^2 + 2\tau - \frac{13}{24}) & 0.75 \leq \tau < 1 \end{cases} \quad with \quad \tau = \frac{t}{T} \tag{5.3}$$

$$p_{goal} = p_{start} + p(\tau) \cdot T^2 \tag{5.4}$$

In the state machine the transitions between states are triggered based on the occurrence of an event or a binary and unary connective of events. Events can be based on time or sensor input (e.g. ground contact), but also on the current state of another state machine, which allows synchronizing multiple states machines.

This offers a more decentralized approach to the control of multiple legs, similar to the neural network control in the Walknet [68] controller, but realized as parallel state machines and with discrete inhibition signals as state transitions. This is used here to generate independent trajectories for the two legs, which are performing similar motions and depending on the desired gate, e.g. alternate or parallel hopping, are actuated with or without phase shift respectively.

In contrast to Raibert's hoppers, which use a completely centralized control with only one state machine, this decentralized control reduces the necessary number of states. A single state machine needs to implement one state for each valid combination of the states of the two parallel machines. For example in [63] it is described that the state machine used for a two legged alternating motion consisted of ten states, where in this approach two synchronized instances of the same state machine with only three states are sufficient as shown in the alternate hopping experiments in Section 6.1.4. The advantage of this approach is that only the gait cycle of one leg and its relation to the other leg have to be described, which leads

to reduced redundancy in the code, fewer errors in the description and implementation and easier adaptation of new trajectory parameters to both legs.

5.2 Joint Level Control

5.2.1 Requirements and Challenges for Joint Level Control of Musculoskeletal Robots

Joint level control of a musculoskeletal human-inspired leg is fundamentally different from that of conventional rigid robotic legs.

First, in conventional robots the rigid coupling of actuator and joint is desired and any elastic deviation from this is seen as an error. In musculoskeletal robots the elastic property of the tendon driven series elastic actuators (TD-SEA) is desired, but introduces a deviation between joint and actuator angles depending on the dynamics of the system and external forces, like the ground reaction forces.

Second, in conventional rigid robot each joint is directly coupled with exactly one actuator possibly via a gear allowing joint control directly by actuator control. The actuator to joint relation in musculoskeletal robots on the other hand is not a one-to-one mapping but rather a mapping of multiple active and passive actuators to each joint. Additionally, via biarticular structures a single actuator can have influence on more than one joint, making joint level control even more complex.

The correlations of actuators and joints in musculoskeletal legs are not yet fully described.

5.2.2 Joint Level Control Approaches

In the BioBiped1 and BioBiped2 robot models used in this work the musculoskeletal actuation design consists only of one active actuator per joint. Ankle and knee joints are each driven by a TD-SEA, which is responsible for the joint extension, while a spring is used as passive antagonist for flexion. In the hip the active actuator is connected to drive the joint in both directions also in a series elastic setup. Through this design the joint control is simplified compared to the aforementioned general requirements for musculoskeletal robots, as only one motor is directly responsible for each joint. But the additional passive biarticular structures still add multi-joint couplings and therefore need to be set up carefully as described in Section 7.3.

Feedback Control

A widely used feedback control approach is proportional derivative tracking control. It can be used to generate control signals based on desired trajectories and the current state of the system without further knowledge about the system dynamics or a model. This allows for a simple setup and fast adaptation to hardware changes. But as any feedback control it operates with a small delay due to its control signal being a reaction to a deviation from the

desired trajectory. So the performance of this type of control is directly dependent on the frequency and latency of the control system, e.g. sensor rate and bus latency. Therefore, requirements on the control system with respect to latency are much higher when using a feedback based control. It also does not take the elastic components of the actuation system with its dynamic properties into account and can therefore not exploit them to improve the robot's performance.

Feed-Forward Control

In a feed-forward control approach the inputs for the actuators are generated using an inverse dynamics model of the robot. This model is used e.g. to compute the motor voltage trajectories needed to track a given desired joint position trajectory. Because this removes the need for feedback information, the requirements on the latency of the control system are lower than for feedback based control. While this makes the feed-forward control acting without any delay, it would need a perfect model to perfectly track a desired trajectory. This model can either be analytically derived or it can be learned from motion data recorded on the robot. Both approaches are tedious and error prone and require adaptation of the model on changes of the robot, even unintentional ones like wear or minimal deformation.

For more details on investigations of tracking performance of a feed-forward control approach based on a learned model see Section 5.3.

Bio-Inspired Control

A control approach using a combination of feedback and feed-forward control is investigated in Section 5.3. It is named bio-inspired control as it uses a feedback rate of only 40 Hz with a 25 ms delay which is comparable to the human nervous system. Due to the lower feedback rate it has reduced requirements on the control system compared to the feedback control. With the feedback component of this approach small errors in the inverse dynamics model can be compensated, reducing the effort needed in the generation and adaptation of this model.

5.3 Model Based Feed-Forward and Bio-Inspired Control

The contents of this section have been previously published in [69].

In conventional robots, elasticity in the actuation is seen as an unwanted property because it increases the complexity of the robot model and its control system. But for musculoskeletal robots, like the BioBiped series, the elasticity plays an essential role in achieving its goal of versatile, energy efficient, dynamic motions. So the problem of modeling and controlling the robot with this increased complexity has to be solved in order to take advantage of its elastic properties.

For a model based control approach an inverse dynamics model of the robot is needed. This model can either be analytically derived or learned based on data generated on the

robot. An analytic model not only has to be manually setup to fit the robot and its interaction with the environment, it also has to be manually adapted to every change of the system (even unwanted changes like wear). Further, to be used as a model for control of dynamic motions the complexity and precision of the model has to be quite high to adequately predict the robot's behavior. Using a learned model solves these two points by directly learning the connection between input and output of the system and relearning the model on system changes. Unfortunately, the learned model produces only good results in the learned parameter regions and cannot easily extrapolate to unknown motions.

The approach shown in this section compares the joint position tracking ability of three different control approaches shown in Figure 5.2. First a naive PD-controller running at 1 kHz with manually selected gains is used, which is sufficient for producing hopping motions as reported in [60]. Based on the data generated with this first approach an inverse dynamics model is learned and used in the other two approaches, namely a pure feed-forward control and a bio-inspired combined feedback/feed-forward control. The bio-inspired control uses a human-like control speed for its feedback component at 40 Hz with 25 ms delay which would be to slow for the control of this system when used without the feed-forward component.

5.3.1 Learned Inverse Dynamics Model

While there are different methods that can be used to learn an inverse dynamics model [46] shows that the non-parametric Gaussian process regression (GPR) is a fast and accurate option for learning complex systems. The model used in the control approaches investigated here is learned using GPR with a Bayesian kernel approach on a training data set recorded on the BioBiped1 robot.

For the Bayesian kernel a squared exponential covariance function is used as covariance matrix

$$k(x,x') = \sigma_f^2 \exp\left[\frac{-(x-x')^2}{2l^2}\right] + \sigma_n^2 \delta(x,x')\,. \tag{5.5}$$

with a constant noise reduction factor σ_n^2. An optimization process is used to maximize the marginal likelihood over the training dataset to generate the hyperparameters for horizontal length-scale l and vertical length-scale σ_f^2.

The training data set is recorded on the BioBiped1 robot using the mentioned feedback controller to generate motions close to the desired motion for the inverse model. The learned model is based on the recorded motor voltages and their correlation to joint positions. A new approach is used to compensate for the missing data about joint velocities and accelerations, which are important for the description of the inverse dynamics. Instead of deriving the missing data from the position and time information multiple time steps are used as additional parameters in each training data set as shown in Figure 5.1. This avoids possible inaccuracies of approximating the data and still gives the model the necessary context for the system dynamics. To account for the complex correlations of multiple joints in the nonlinear dynamics

of this musculoskeletal robot, the training set for each joint also includes the data from all other joints.

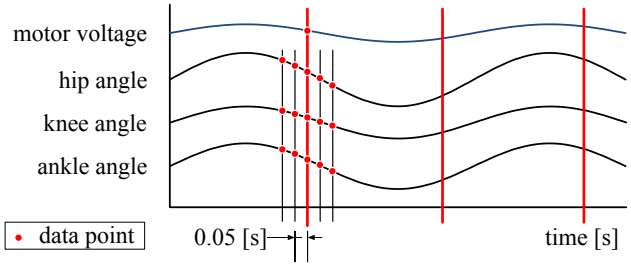


Figure 5.1: Schematic visualization of the 16 data points (red dots) contained in one training data sample for one of the motors.

Published in [69]

5.3.2 Feed-Forward Control

The learned model can now be used to convert a desired joint trajectory into motor input voltages which can be used in a feed-forward controller as depicted in Figure 5.2. To ensure good results using the model the desired trajectory should be close to the ones used in the training data. For a successful feed-forward control, the robot's initial state needs to be as close to the start position of the desired trajectory as possible. This is achieved by starting with the feedback controller to move into this position and then switch to the feed-forward control.

5.3.3 Bio-Inspired Control

The bio-inspired control approach evaluated here uses a similar setup to the feed-forward approach, but adds a feedback control component to gain the advantages of both approaches as shown in Figure 5.2. It is named bio-inspired since the feedback control runs with a slow feedback loop at 40 Hz and a 25 ms delay - the approximate speed of the human nervous system. Also, the feedback gains are reduced to 10% of the values used in pure feedback control. Again the robot is brought into its initial position using the feedback controller before switching to the bio-inspired control.

5.3.4 Experiments

To compare the three control approaches they were each tested using two different motions, one without and one with ground contact.

In the first motion the robot's trunk is attached to an external frame, holding the robot in the air with its legs hanging as can be seen in Figure 5.3 on the left. This allows to first test the control approaches against a motion without the influences of ground contact. Here one leg

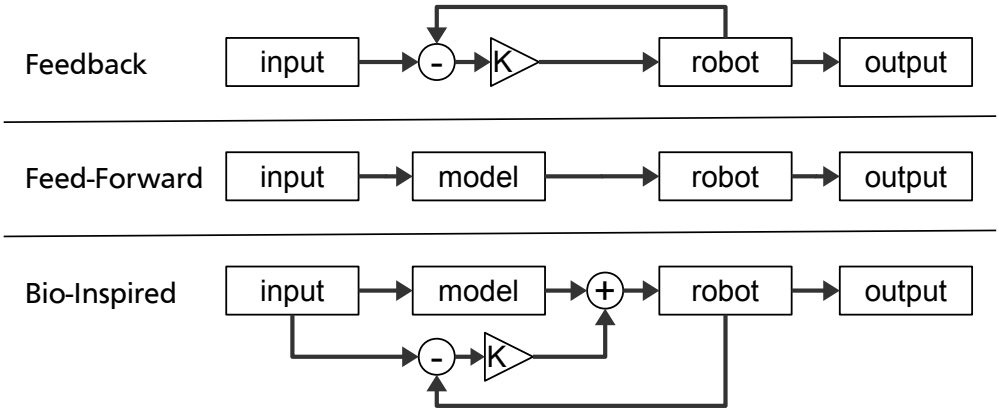


Figure 5.2: Schematic control diagrams of the feedback, feed-forward and bio-inspired controllers. The input block produces the desired joint positions, while the robot block outputs the actual joint positions. The input to the robot block coming from the different controllers are the motor voltages.

Source: own representation

was to perform the transition between the retracted and touchdown positions of the alternate hopping motion described in 6.1.4.

The second motion is performed with the robot standing with both foot tips on the ground supporting its own weight and its trunk constrained to vertical translation by an external frame as can be seen in Figure 5.3 on the right. Here the desired motion is a periodic up and down swinging motion used in the synchronous hopping experiment before liftoff (see 6.1.3).

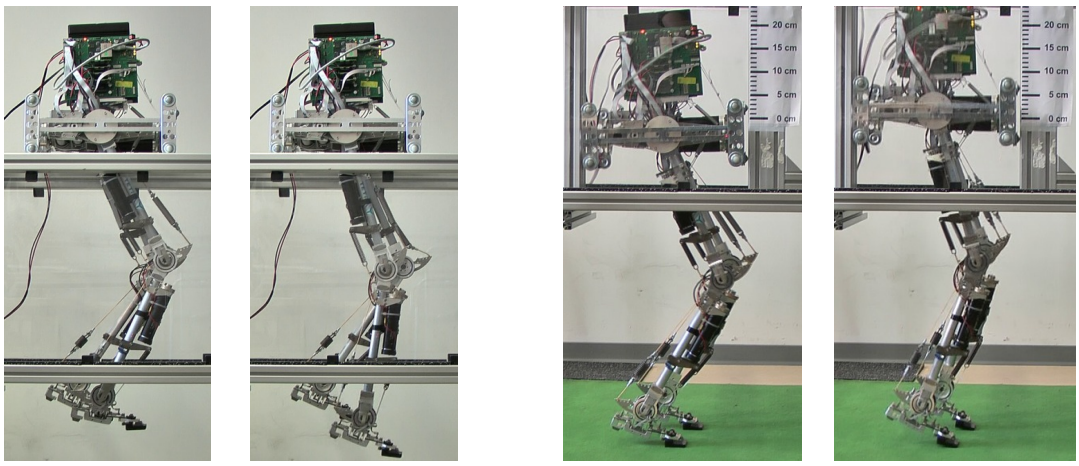


Figure 5.3: Experiments 1 (left) with the robot hanging from its trunk moving one leg between retracted and extended positions. Experiment 2 (right) with the robot standing on its foot tips and alternating between bent and extended positions.

Source: own representation

5.3.5 Results

As can be seen from the trajectory plots shown in Figure 5.4 and Figure 5.5 all three controllers showed a reasonable performance for both motions. The PD-controller exhibits a small delay in the tracking of the target trajectory for both motions. In the first motion an increasing oscillation especially in the hip joint can also be seen, while in the second motion an overshot in the hip trajectory is can be clearly seen. This is the expected behavior of such a simple controller applied to a complex elastic system.

While the feed-forward approach does not show these problems, it drifts away from the target over time, best seen in the knee joint in Figure 5.4 and the ankle joint in Figure 5.5.

The bio-inspired control combines the advantages of the other two controllers which can be seen in the two plots in Figure 5.6. Here the joint position errors in each joint for all recorded data points are shown. It clearly shows the bio-inspired approach has the smallest position error in all joints in both motions by reducing both, the typical delay of the feedback control and the drift of pure feed-forward control.

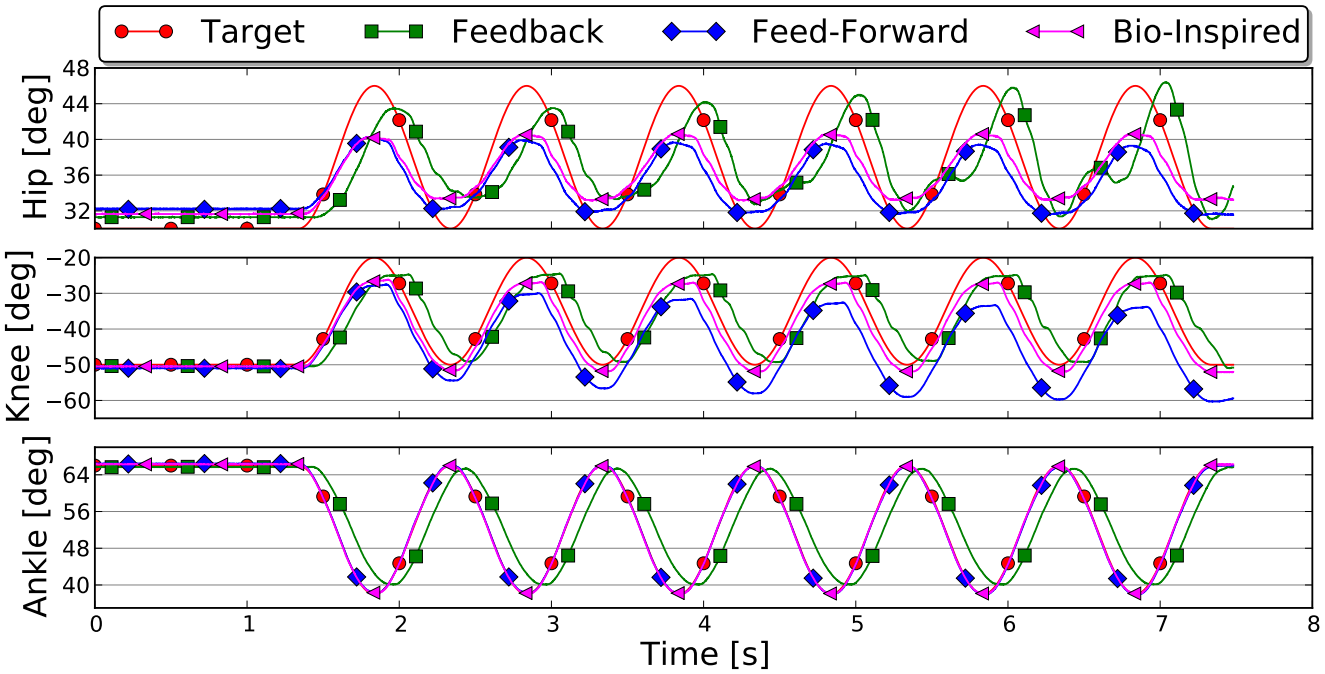


Figure 5.4: Experiment 1: Target trajectories and resulting joint angle trajectories for feedback, feed-forward and bio-inspired control.

Published in [69]

5.3.6 Conclusion

Three joint level control approaches were investigated for the control of the highly elastic musculoskeletal biped robot BioBiped1.

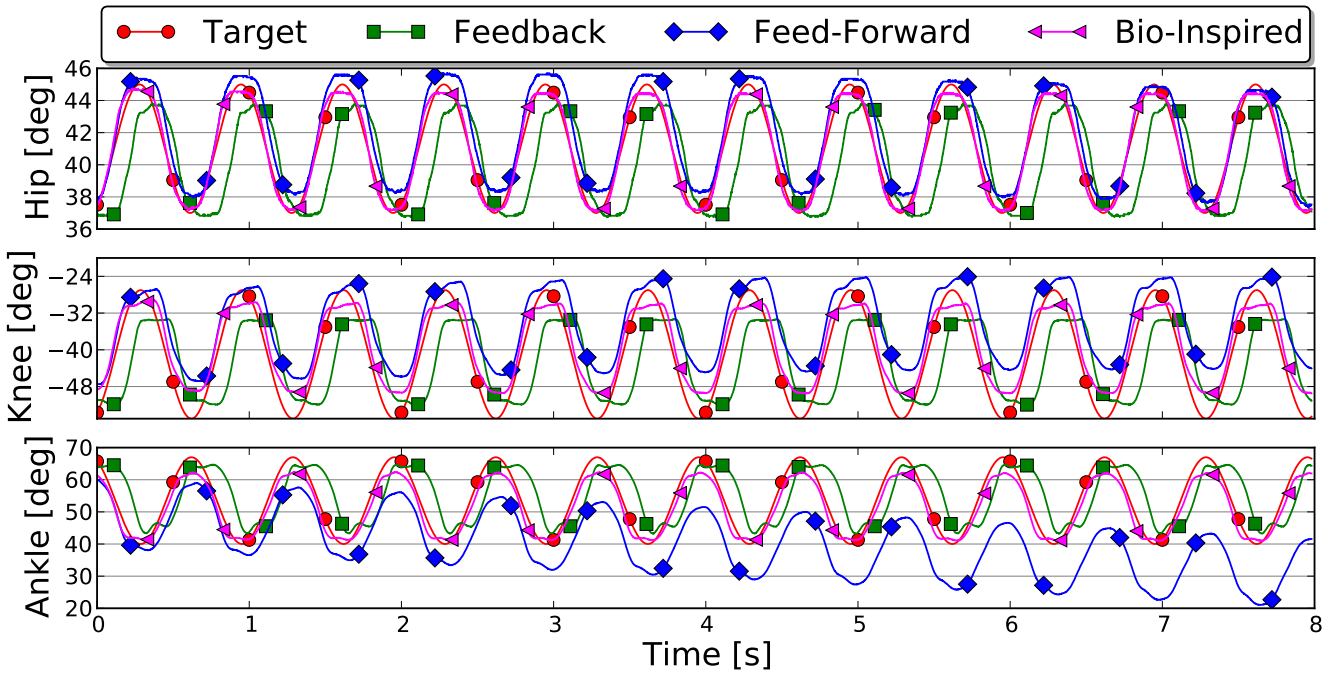
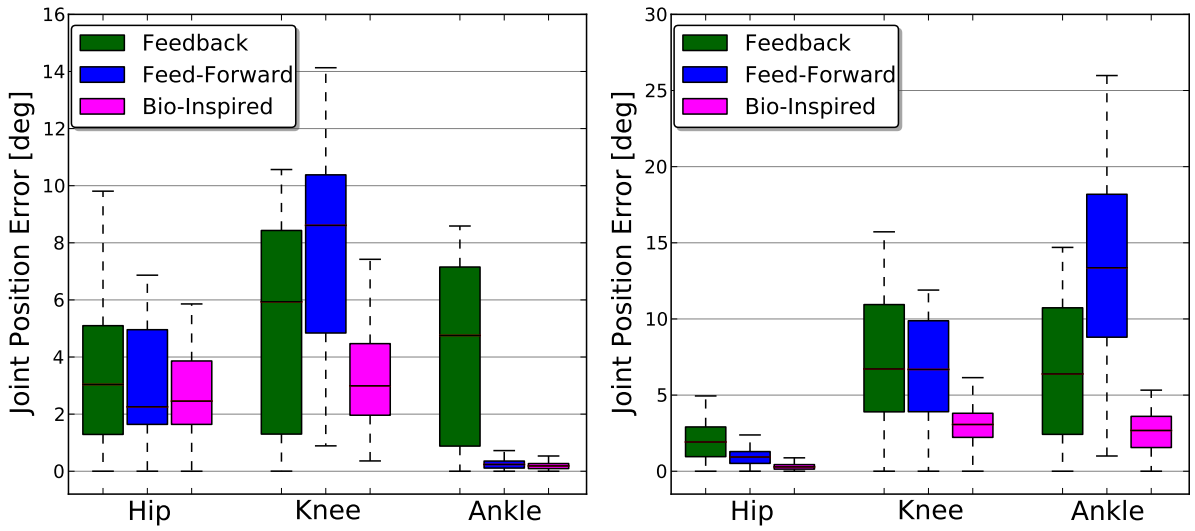


Figure 5.5: Experiment 2: Target trajectories and resulting joint angle trajectories for feedback, feed-forward and bio-inspired control.

Published in [69]



(a) Experiment 1

(b) Experiment 2

Figure 5.6: Joint position error for feedback, feed-forward and bio-inspired control for the two experiments.

Published in [69]

A naive feedback controller which is able to successfully produce different hopping motions as described in Chapter 6 was chosen as baseline for comparison. While it has low implementation effort, it does not however account for the correlations of the dynamics of

multiple joints and the series elastic actuation system and therefore shows the typical symptoms of delayed tracking, overshooting and oscillations.

The second candidate, a pure feed-forward multi-variable controller based on a learned inverse dynamics model, does not show these problems. But as it is missing feedback from the system, any difference between learned model and system leads, e.g. minimal deviation in the initial condition, lead to a tracking error growing over time.

The third approach combines the model based feed-forward control with a slow delayed feedback control to a bio-inspired controller. This leverages the advantages of the first two by reducing the delay, oscillations and overshot with the feed-forward component and compensating the growing tracking error with the feedback part.

While using a learned model as part of the control showed to be valid approach, it needs significant effort to keep the learned model updated with all (unwanted) changes in the system. Therefore, an automatic self updating model would make this a viable approach for further use, but has not been investigated further in this work.

Furthermore, feed-forward control is used in humans for fast motions like running. Here the goal of the human control system is not to track individual joint trajectories, but rather to achieve a global motion goal, e.g. foot positioning. So a next step would be to evaluate the bio-inspired control approach when applied to such motions goals.

6 Experimental Evaluation of Basic Functionality

Prior to the development and adaption of suitable passive (i.e. spring stiffnesses, attachment points) and active control parameters in the systematically guided experiments investigated in Chapter 7, it is an important first step to test and evaluate a number of basic functionalities of the system. A set of experiments was designed to cover the following functionalities:

- Mechanical robustness of the system
- Energy restitution of the elastic leg
- Actuation system dimensioning
- Exploitation of the system's eigenfrequency
- Robustness against perturbations

Each functionality is covered by one or more of the experiments as can be seen in Table 6.1. The experiments were conducted in increasing order of mechanical demand on the robot to be able to correct possible problems before they lead to serious damage of the robot. Some figures shown in this chapter have already been published in [60] and are marked accordingly.

	Passive Rebound	Single Push-Off	Synchronous Hopping	Alternate Hopping	Perturbed Hopping
Mechanical robustness	x	x	x	x	x
Energy restitution	x		x	x	x
Actuation system dimensioning		x	x	x	x
Eigenfrequency exploitation			x	x	x
Robustness of motions			x	x	x

Table 6.1: The coverage of the evaluation of basic functionalities through the experiments.

During all the experiments shown here the software components described in 4.5 were used to control the robot, monitor its state, record all sensor data and analyze them. The author of this thesis has implemented all the mentioned software components, operated the robot control software during the experiments and evaluated the recorded data.

The ability to monitor all data during live operation was important for manual tuning of parameters and to avoid damages to the system due to overheating in the actuators or overextension of springs.

Off-line analysis of the recorded data between experiments is an important part of the experiments, as most parameters in the experiments in this chapter were tuned manually. Therefore, an automated data filtering and visualization process helps to speed up the analysis time needed in between experiments, leading to a more time efficient operation.

	Passive Rebound	Single Push-Off	Synchronous Hopping	Alternate Hopping	Perturbed Hopping
VAS stiffness [N/mm]	7.9	15.5	15.5	15.5	15.5
VAS knee lever arm [mm]	66.5	50.8	43.0	58.7	58.7
SOL stiffness [N/mm]	6.7	7.9	13	13	13
SOL ankle lever arm [mm]	50.8	50.8	43.0	58.7	58.7
GAS stiffness [N/mm]	4.1	4.1	-	-	-
GAS knee lever arm [mm]	32.6	32.6	-	-	-
GAS ankle lever arm [mm]	66.5	36.4	-	-	-
PL/TA stiffness [N/mm]	rope	rope	4.1	4.1	4.1

Table 6.2: Setup of the adjustable hardware parameters used in the different experiments.

6.1 Description of Experiments

In this section the setup and execution of the experiments which were designed to evaluate the aforementioned basic functionalities of the robot are described. The evaluation of the results follows in Chapter 6.2.

6.1.1 Passive Rebound

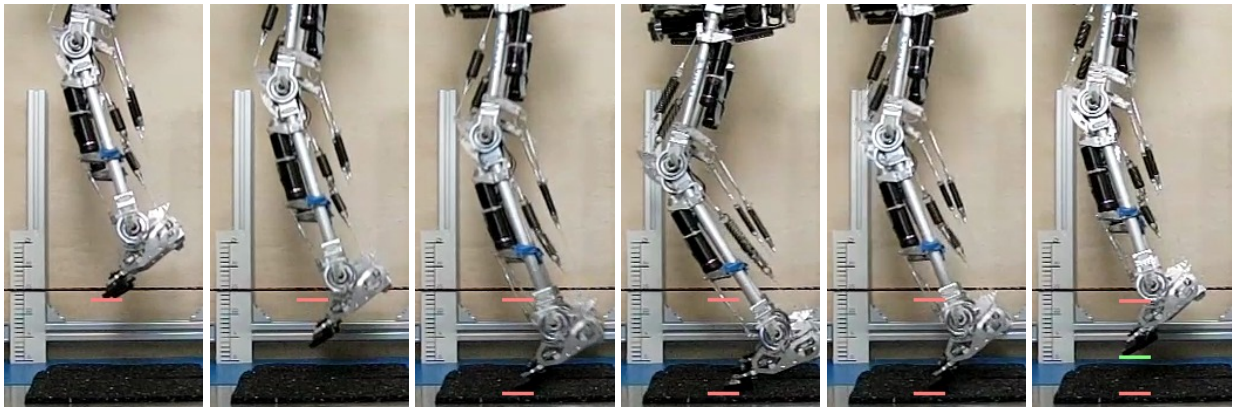
The first experiment performed with the newly developed robot prototype was a passive rebound after a drop from a fixed height to test its mechanical robustness and quantify the energy restitution of its legs. The motors were controlled by a PD-controller to stay in fixed positions during the whole experiment. To keep the robot in the desired initial position while it was held in the air, ropes were used as flexors for knee and ankle. Their lengths were adjusted to keep the joints in their initial positions at 140 degrees for the ankle and 155 degrees for the knee.

The robot was dropped without any external support while being recorded by a high speed camera to measure the rebound height (Figure 6.1). The rebound occurred on a force plate measuring the ground reaction forces (Figure 6.2) for use as ground truth data to calibrate the robot's foot force sensors. After the second rebound the robot needed to be externally stabilized to prevent a fall, so only data recorded before this are evaluated.

Multiple configurations for the springs used in VAS, SOL and GAS were tested leading to the most successful parameters shown in Table 6.2.

6.1.2 Single Push-Off

To test the ability of the robot's actuation system to power a push-off motion needed to initiate the following hopping experiments, first a single two-legged push-off was performed. The robot was put on a force plate in a bent standing position with the knees at 125 degrees and the ankles at 95 degrees. After about 100 ms of standing without external support the goal position for the PD-controllers was switched to an extended leg configuration thrusting the



(a) apex (b) apex (c) falling (d) touchdown (e) compression (f) liftoff

Figure 6.1: Passive rebound from 15 cm drop.

Published in [60]

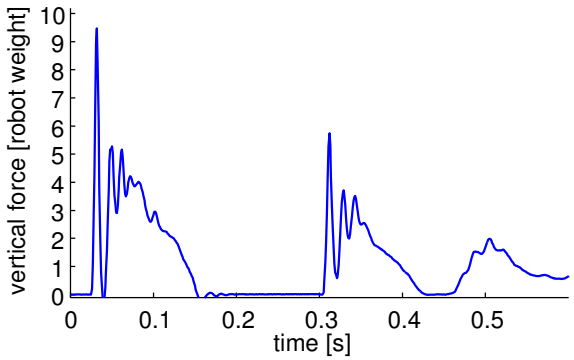


Figure 6.2: Vertical ground reactions force for passive rebound normalized to robot weight.

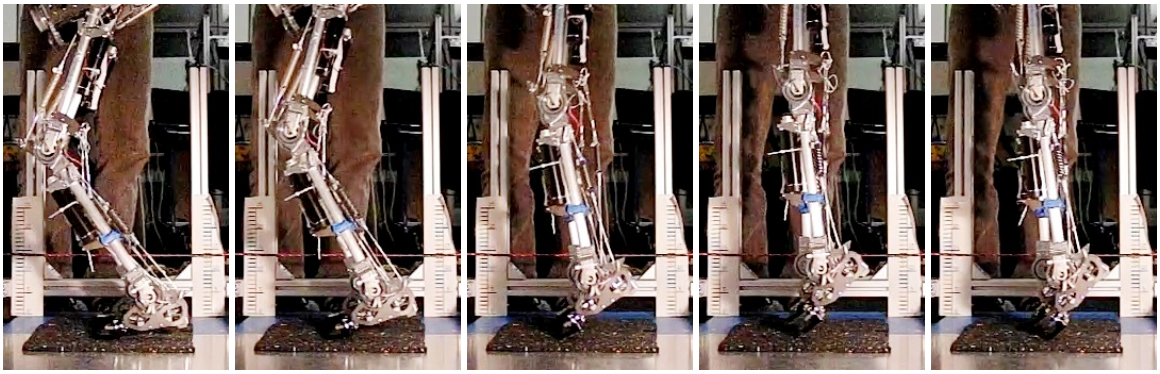
Published in [60]

robot upwards. Again the robot needed to be externally stabilized after the first flight phase to prevent it from tilting over.

The setup parameters of the robot can be found in Table 6.2.

6.1.3 Synchronous Hopping

The robot is hopping with a synchronous motion of both legs while its trunk is externally constrained to vertical translation. The desired motor trajectories are generated by switching both legs simultaneously between two different configurations corresponding to a bent and an extended position represented by goal positions for the hip, knee and ankle extensor motors. To transition between these two goal positions either a direct switching or a smooth transition of the desired position for the PD-controller can be used. For the smooth transition a trajectory is interpolated starting from the current motor position and ending at the goal position using the function described in Section 5.1.2.



(a) initial position (b) extending (c) liftoff (d) apex (e) touchdown

Figure 6.3: Single push-off from bent position.

Source: own representation

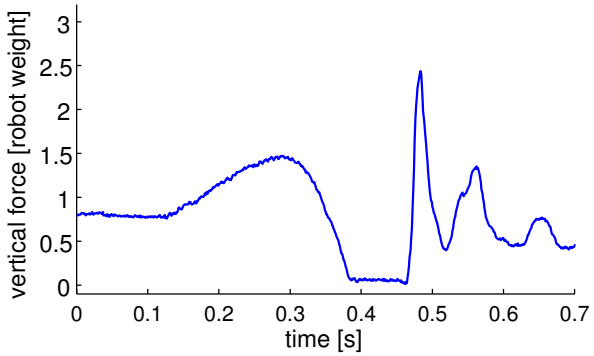


Figure 6.4: Vertical ground reactions force for single push-off normalized to robot weight.

Published in [60]

Switching the positions is realized using a single state machine (Section 5.1.2) for both legs with two states shown in Figure 6.5, each representing one set of desired motor angles. The transitions between these two states can either be based on fixed timing or can be initiated by the detection of touchdown and liftoff from the forefoot ground contact force sensors. While the fixed timings for the transitions can be tuned to allow for continuous hopping its does not always leverage the eigenfrequency of the system. The eigenfrequency of the system depends on the overall leg stiffness, which depends on the selected spring stiffnesses at each joint, but also the joint angles during ground contact. Since the spring selection can be changed in between experiments and the landing configuration can vary even between individual hops a fixed timing does not always fit perfectly. So incorporating the transitions based on the ground contact detection allows adjusting the timings dynamically. But the fixed timings are still necessary in the beginning of the motion to allow the robot to start with an up and down motion (while staying on the ground) to build up the energy in its elastic elements needed for liftoff.

The setup parameters of the robot can be found in Table 6.2.

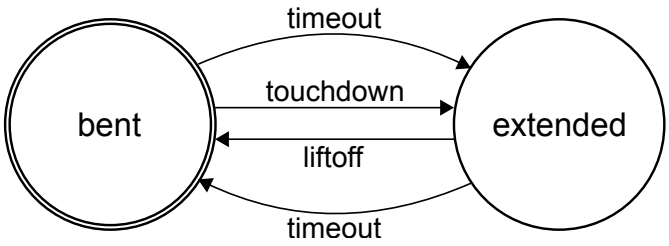


Figure 6.5: State machine for synchronous hopping with transitions based on timeouts or ground contact events.

Source: own representation

First Approach

In the first approach a feed-forward generation of the motor trajectories and a direct switching between the goal positions for bent and extended configurations was used. Switching between the two states representing the leg configurations was only done based on time and had therefore be tuned to fit the current elastic properties of the legs. The resulting trajectories can be seen in Figure 6.6.

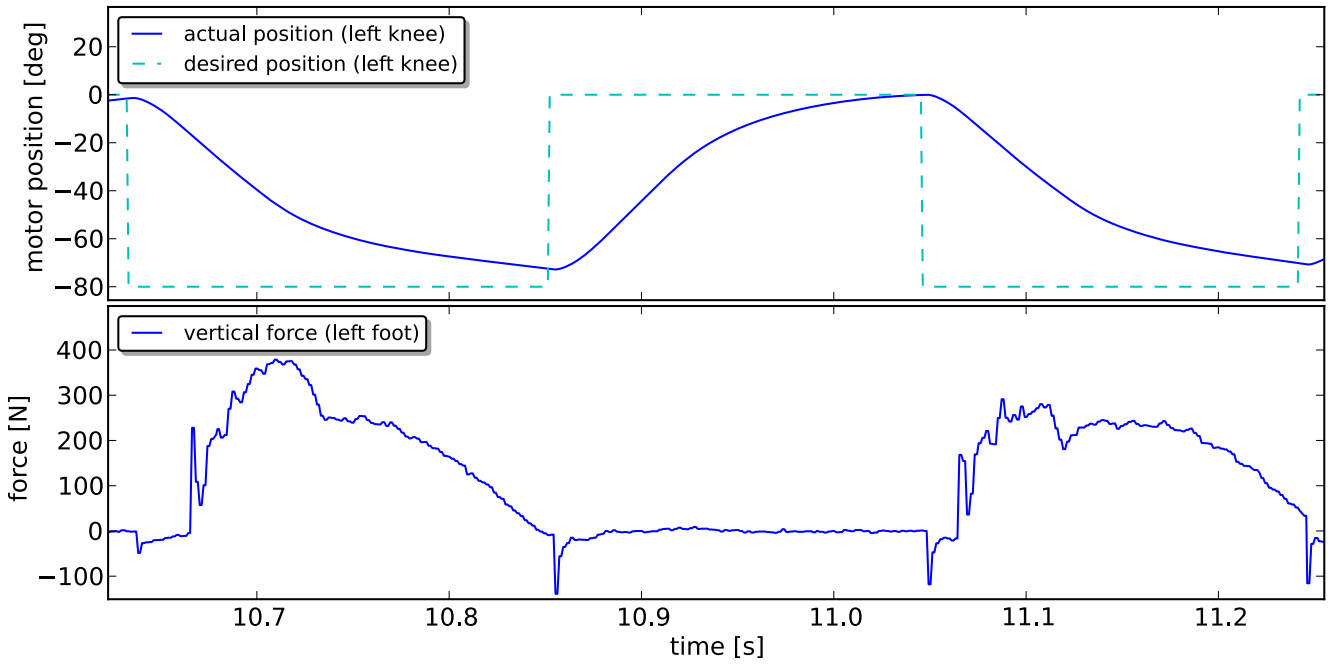


Figure 6.6: Synchronous feed-forward hopping. Top: Goal and actual position of the knee motor. Bottom: Vertical ground reactions force measured at the foot tip.

Source: own representation

Second Approach

In the second approach the feedback from the foot-force sensors was used to detect touchdown and liftoff events. Based on these events the states for bent and extended configurations were switched, allowing the trajectory to be adapted to the current elastic properties of the legs. Additionally, the switching between goal positions was smoothed by interpolation as detailed in Section 5.1.2 to produce less wear in the motors and gears. The resulting trajectories can be seen in Figure 6.7.

6.1.4 Alternate Hopping

The robot is hopping alternately on its left and right leg while the trunk is again externally constrained to vertical translation. The motion generation is done similar to the previous experiment of synchronous hopping, but with two instances of the state machine, one for each leg and each comprising three states (Figure 6.8). The states are again one for bent, one for extended and a new one for retracted leg configurations. Further the state machines' transitions are extended to synchronize the two state machine instances, forcing the two legs to alternate in ground contact (Figure 6.9). Again a combination of fixed timings and ground contact detection can be used to trigger state transitions.

Two motion generation approaches were tested, analog to the ones described in the synchronous hopping experiment. First with timing based feed-forward generation of a trajectory and direct switching of the goal angles (Figure 6.10). And second with additional feedback from ground contact events for state switching and smoothed transitions between goal positions.

6.1.5 Perturbed Hopping

This experiment is based on the synchronous hopping motion with ground contact feedback and smoothed goal trajectories. While hopping the robot is externally perturbed by changing the ground height and applying horizontal forces to the legs while in flight to change the touchdown positions. To alternate the ground height in between touchdowns two triangular shaped wedges were inserted or removed under the left or right ground plates the robot was hopping on. This resulted in changes in the ground height of up to 3 cm for each leg individually.

Further, with ropes attached to the front and back of the legs horizontal forces were manually applied during the flight phase leading to horizontal variations in the landing positions of the feet of up to 10 cm for each foot individually.

In Figure 6.11 photos of three exemplary landing configurations and the resulting liftoffs can be seen. Shown are a touchdown and liftoff for an unperturbed hop, for a hop with changed ground height for the left leg and for a hop with the right leg pulled forward.

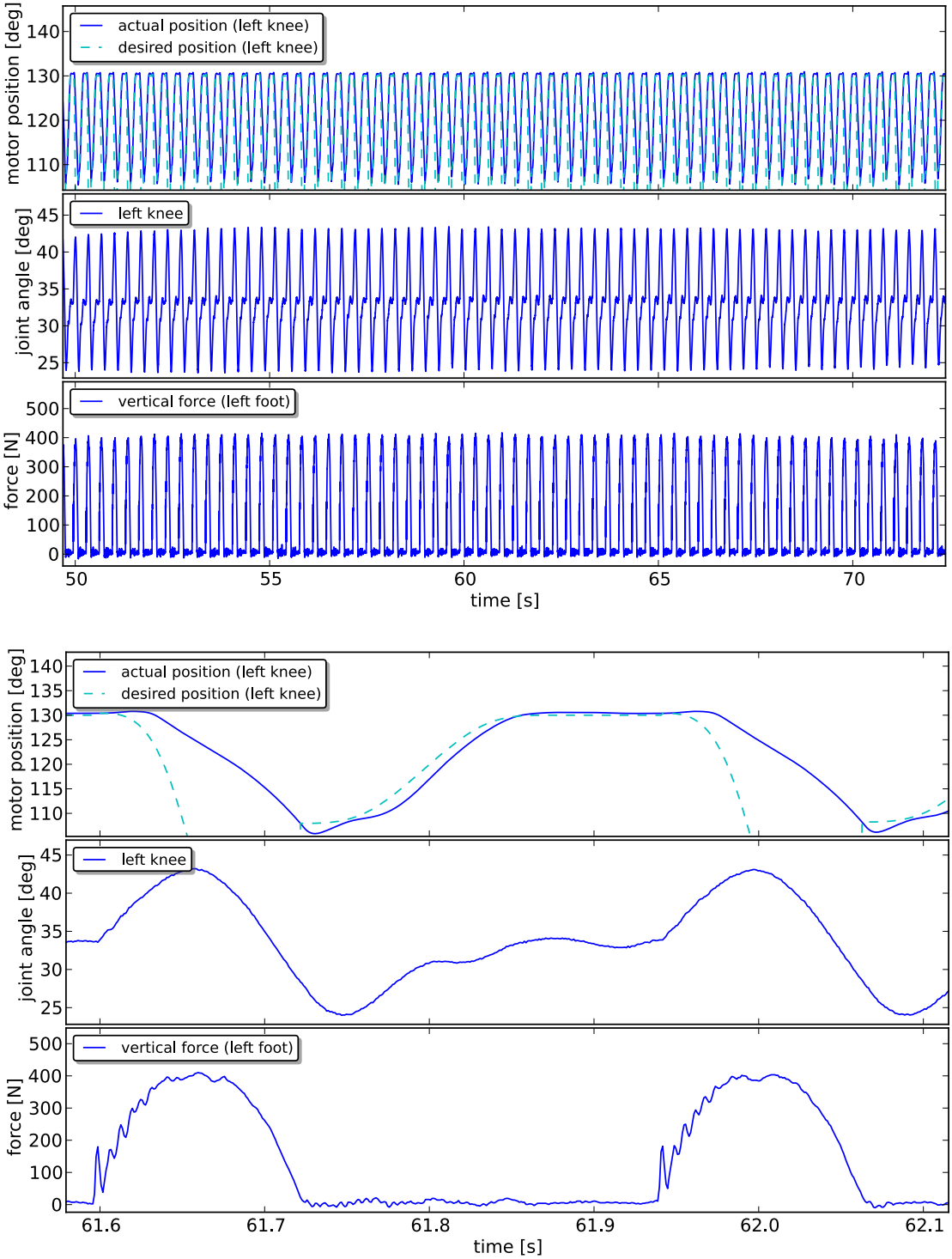


Figure 6.7: Data recorded during synchronous feedback hopping showing the robustness of the motion for more than 60 hops in the top figure and the details of one hop in the bottom figure. The data shown contains the desired and actual positions of the left knee motor, the left knee joint angle and the vertical ground reactions force measured at the left foot tip. The hop shown in the bottom figure has a cycle time of 343 ms with a duty factor of 0.37.

Source: own representation

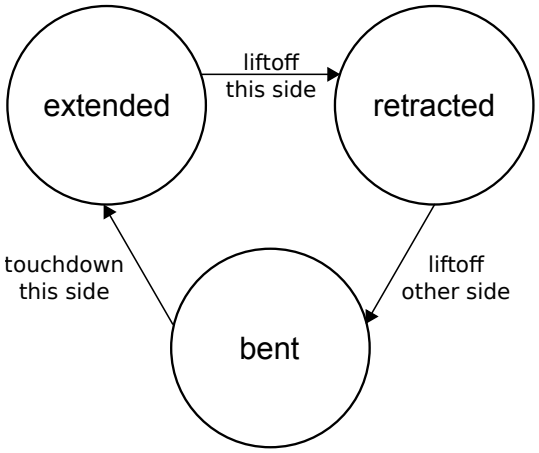


Figure 6.8: To generate an alternate hopping trajectory two instances of this state machine are running in parallel (one for each leg) with transitions based on ground contact events of both legs. The motion is started by dropping the robot with one state machine in 'bent' state and the other in 'retracted'.

Source: own representation

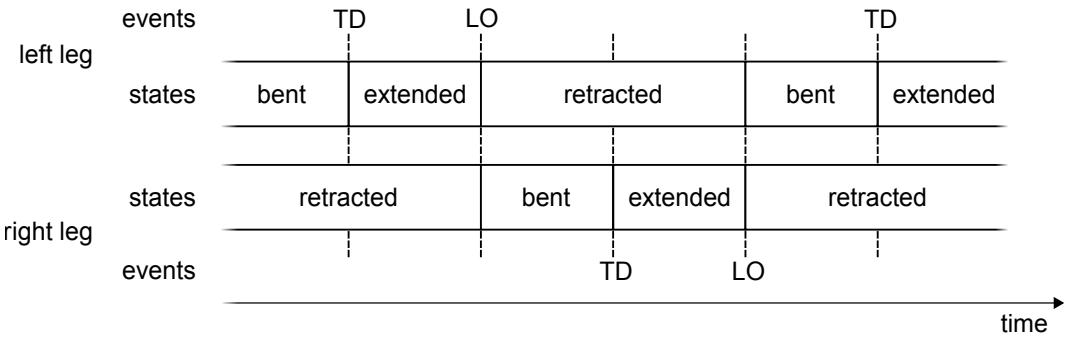


Figure 6.9: Timing diagram of the two state machines producing an alternate hopping trajectory for two legs based on touchdown (TD) and liftoff (LO) events from both feet.

Source: own representation

An excerpt of the resulting trajectories can be seen in Figure 6.12, showing the time around the landing with the right foot pulled forward seen in Figure 6.11(e). The altered landing position leads to changed patterns in the ground reaction forces and the knee joint trajectory for one cycle. As can be seen from the two following hops they quickly return to the previous patterns in both joint angle trajectory and ground contact forces.

6.2 Evaluation of Results

6.2.1 Mechanical Robustness of the System

The first evaluation of the mechanical robustness was done in the passive rebound and single push-off experiments. In both cases the robot showed no signs of mechanical damage or deformation. Based on this the robot was seen fit to perform more demanding motions with

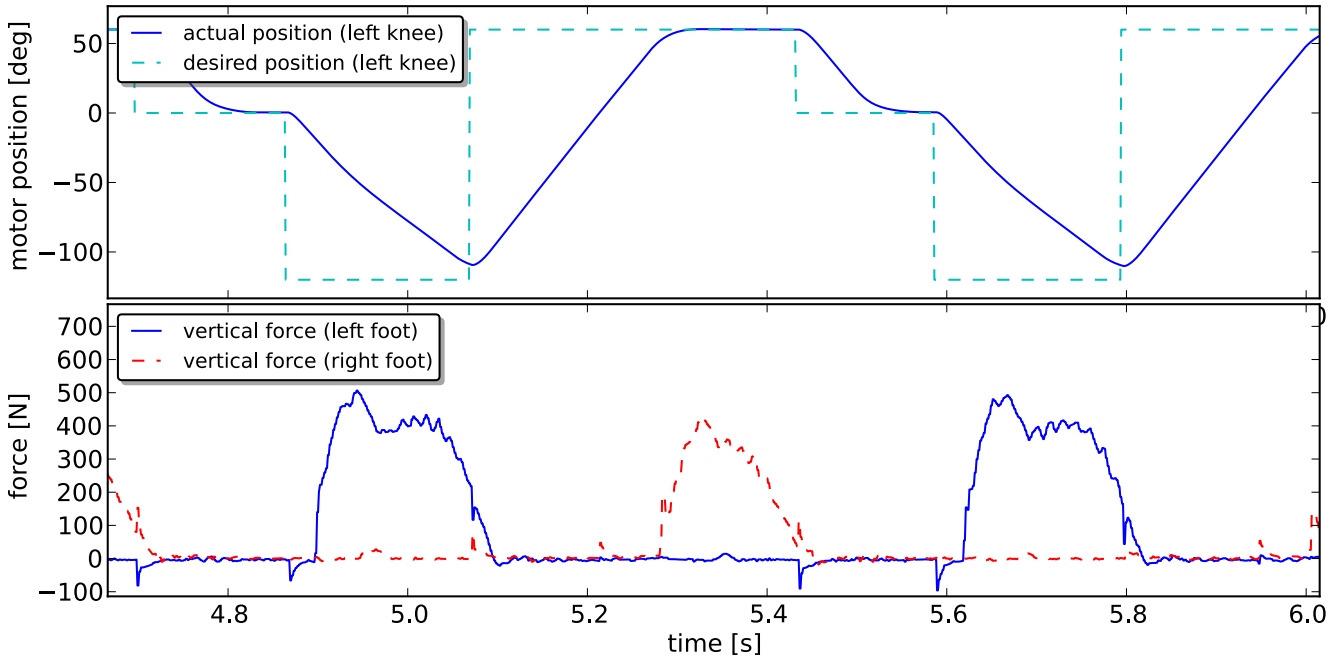
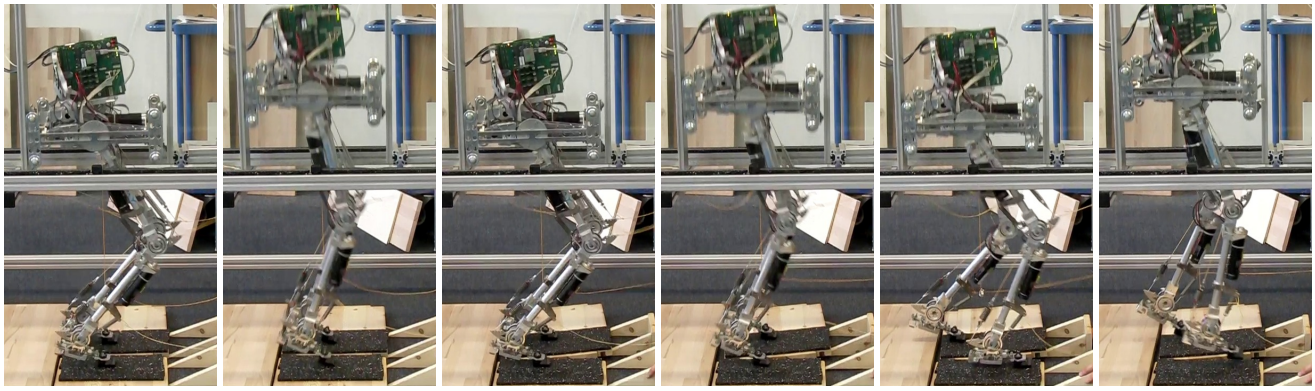


Figure 6.10: Alternate feed-forward hopping. Top: Goal and actual position of the knee motor. Bottom: Vertical ground reactions force measured at the foot tip.

Source: own representation



(a) compression (b) liftoff (c) compression (d) liftoff (e) compression (f) liftoff

Figure 6.11: Synchronous hopping with external perturbation through changes in ground height and horizontal forces applied to the legs. Three compression phases are shown: (a) unperturbed landing, (c) landing with changed ground height for the left leg and (e) landing after the right leg has been pulled forward (time 88.25 s in Figure 6.12). The corresponding liftoffs are shown in (b), (d) and (f).

Source: own representation

continuous hopping. In both the synchronous and alternate hopping experiments the robot was able perform more than 100 continuous hops, limited mostly by mechanical wear of the actuating ropes. In BioBiped1 the ropes were guided around small pulleys on plain bearings,

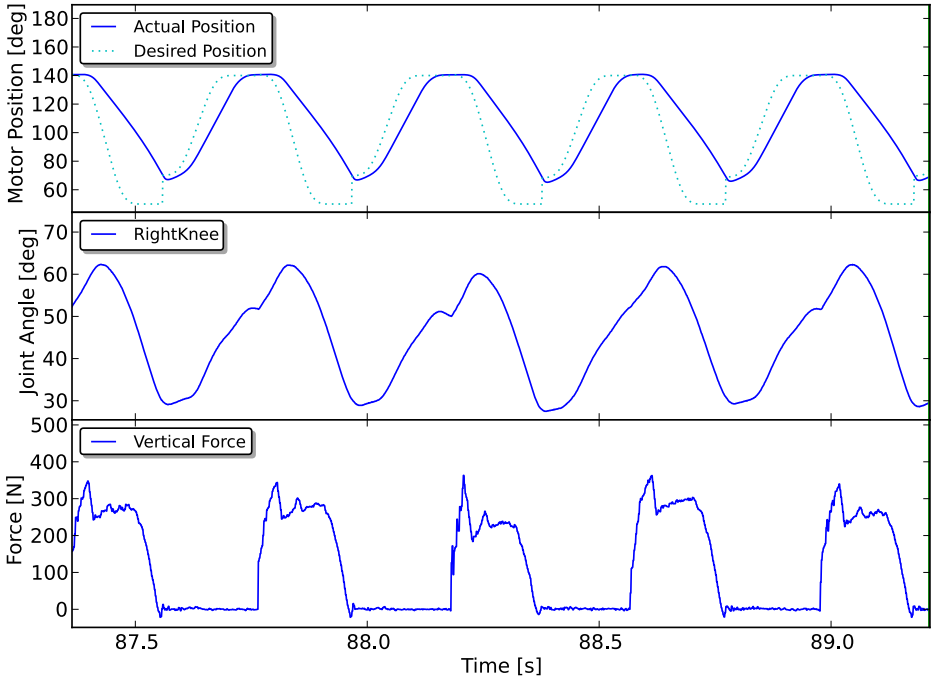


Figure 6.12: Data recorded for the right leg during synchronous hopping with external perturbations and ground contact feedback. Before the touchdown at time 88.2 s, this leg was pulled forward as shown in the photo in Figure 6.11(e). This leads to a change in the ground reaction force (GRF) of this hop (bottom) and different joint trajectory in the next flight phase (middle). The motor trajectories are not affected (top). Both GRF and joint trajectory return to their previous patterns in the following hopping cycle.

Source: own representation

which lead to quick wear of the ropes and some heating of the ropes due to friction. This was addressed in an updated version of BioBiped1 and later generations by larger pulley on ball bearings. Nevertheless, the ability to perform these hopping gaits is a good indicator for the physical ability of the system to perform at least a slow jogging motion. Over longer series of experiments the BioBiped1 robot showed some degradation in the hip roll joints' and the body pitch joint's mechanics leading to backlash in the actuation of those joints. For the BioBiped2 robot those joints were removed to gain mechanical robustness. Further the friction of the plain joint bearings in ankle and knee joints of BioBiped1 increased and also became angle depended due to wear. To overcome this issue in BioBiped2 they were replaced by ball bearings. More details about the mechanical improvements made for the robot generations can be found in Section 4.6.

6.2.2 Energy Restitution of the Elastic Leg

The vertical component of the normalized ground reaction forces recorded by the force plate during the passive rebound experiment can be seen in Figure 6.2. It shows that the first rebound results in a flight phase lasting about 150 ms and a second rebound. The height of

the first rebound was measured from the high speed camera images to be 5 cm after a 15 cm drop, which confirms an energy restitution of the leg elastic structure of about one third.

The energy restitution of the elastic elements can also be seen when comparing the results from the single push-off and the synchronous hopping experiments. In both experiments the same push-off motions was used, but it resulted in a much longer flight phase of 200 ms in the synchronous hopping compared to the 80 ms in the single push-off. This improved performance can be attributed to the energy storage in the elastic elements in the repeating motion compared to the single shot motion.

6.2.3 Actuation System Dimensioning

In the single active push-off experiment ground reaction forces (see Figure 6.4) show an increase during the push-off followed by a flight phase of about 80 ms. This shows the actuation system's ability to generate enough thrust to lift the robot off the ground, which is necessary for hopping experiments.

In the plot of the synchronous hopping experiment (see Figure 6.6) it can be seen, that the motors have enough power to move against the forces induced by the touchdown. Here the knee motor can be seen moving towards the extended position over the whole stance phase, first extending the spring and then also extending the joints again leading to the next liftoff.

The same holds for the alternate hopping (see Figure 6.10) were the motors of a single leg have to do all the work. Even though the motor power was high enough for single legged push-off, the overall transmission ratio in BioBiped1 was too high for fast retraction of the swing leg. This was addressed by hardware changes in BioBiped2 detailed in Section 4.6.

6.2.4 Exploitation of the System's Eigenfrequency

By using the ground contact events instead of fixed timings to switch between the extended and bent leg configurations during synchronous hopping the hopping performance of the system was increased. This can be attributed to better exploitation of the system's eigenfrequency which can change between multiple hops due to different landing configurations and is therefore not optimally suited by any fixed value. The duty factor of the robot was reduced from 0.47 using fixed timings (see Figure 6.6) to 0.37 using ground contact event based switching (see Figure 6.7).

6.2.5 Robustness of Motions

The synchronous and alternate hopping motions were robust enough to perform more than 100 continuous hops. Limiting factor was only mechanical wear in the ropes as described earlier. The passive elastic properties of the system make it robust against variations in ground contact timing so that even with the first approach using fixed cycle times hopping is possible, even though the fixed timings do not fit exactly each hopping cycle. Adapting the

cycle times based on the ground contact feedback increased the performance of the hopping as described above for the duty factors. But it also made the motions more robust, even against manual external disturbances in ground contact time and position as described in Section 6.1.5. Without any need to adapt the trajectory generation or motor control, hopping on changing ground heights and with manually altered landing positions of the feet is possible. This can be attributed to the system's inherent mechanical adaptation to the external disturbances through its elastic elements and validates that the design approach used in the BioBiped robot series helps to perform robust locomotion.

7 Expert Guided Hardware-in-the-Loop Motion Optimization for Musculoskeletal Bipedal Robots

In a hardware-in-the-loop optimization, experiments are conducted on the robot hardware to determine the value of a quality criterion which is to be optimized. This chapter first motivates the use of hardware-in-the-loop optimization in comparison to optimizing solely using a simulation model. Then other approaches to optimization of bipedal musculoskeletal robots are described. An example application of an established hardware-in-the-loop optimization approach for the BioBiped1 robot and its shortcomings are discussed. As the increased complexity of a musculoskeletal robot with its mono- and biarticular elastic structures offers an even larger number of variables to optimize compared to conventional stiff robots, a direct hardware-in-the-loop optimization would need more hardware experiments than practically feasible. Therefore, a new concept for hardware-in-the-loop optimization is presented, integrating expert knowledge and simulation results into the optimization process to reduce the number of hardware experiments needed. This new concept is then demonstrated in an example optimization of the performance of a synchronous hopping motion for the musculoskeletal BioBiped2 robot.

7.1 Motivation and Problem Formulation

Hardware-in-the-loop optimization, where the robot hardware is used to evaluate a quality criterion, plays a very important role in the optimization of mechanical and control parameters of musculoskeletal robots. Even though optimization using a simulation model has a much smaller cost per experiment, the hardware is its only perfect "model". A simulation model is always subject to idealization and abstraction only reflecting the properties of the real robot that were deemed relevant for the goal of the simulation. And there will always be a gap between simulations and real world systems as not all details can be identically mapped to a model for simulation [32]. Dynamic effects that are very difficult to model perfectly include changes in contact with the ground or a constraining mechanism, internal friction and spring properties of physical linear springs, which are never completely linear. Therefore, the limited accuracy of current simulation models for highly dynamic motions with musculoskeletal robots still does not allow fine tuning parameters for direct use on the robot hardware. Furthermore, the robot hardware does not stay exactly the same over its whole lifetime as it is affected by longterm effects like wear and short term effects like changes in temperature or humidity, which are rarely accounted for in any practically usable simulation model of a complex robotic system. A simulation model is rather built with idealizations to be practically usable in terms of complexity and run-time performance by modeling only the most relevant properties while still fulfilling a specific purpose. Nevertheless, a good simulation model can be used to identify and exclude parameter spaces that will not give good results when applied to the robot or might even damage the system. Using the knowledge

gained from these simulation experiments to systematically plan the robotic experiments helps to reduce the number of experiments needed on the robot thereby preventing potential damage and reducing the wear inflicted upon the hardware as well as the time needed to perform the experiments.

Using a hardware-in-the-loop optimization approach with a robot can pose certain additional requirements on the robotic system. In order to be able to evaluate a desired optimization criterion directly using the robot, it might be necessary to add specific sensors to the system, which would otherwise not be needed in the normal operation of the system.

The parameters that are to be optimized range from mechanical passive control parameters like spring stiffnesses and lever arm lengths to active control parameters like controller gains or trajectory parameters. Because of the large number of these parameters in a musculoskeletal robot, a pure black box optimization conducted on the hardware would need more robot experiments than are practically feasible. A new concept is therefore presented in Section 7.3 to reduce the number of experiments needed in an expert guided hardware-in-the-loop optimization by the application of structured information from simulation experiments, biomechanical understanding of the system and knowledge from previous experiments. But first a conventional hardware-in-the-loop approach is discussed in the following section.

7.2 Conventional Approach of Hardware-in-the-Loop Optimization applied to BioBiped1

The contents of this section have been previously published in [70].

During bouncy gaits like fore-foot running or hopping the human leg is loaded and unloaded during ground contact while pivoting around the fore-foot [16]. Human-like three segmented legs as used in the musculoskeletal BioBiped robot series offer the versatility to perform similar motions. But in combination with series elastic joint actuation they also bear the risk of overextension in the knee or ankle joint leading to a bow leg configuration instead of the desired zig-zag configuration [73]. In bouncy gaits this risk occurs during the loading and unloading phases and is subject to the stiffness ratios and initial positions of the concerned joints. In the human leg the Gastrocnemius muscle (GAS) helps to prevent this overextension by synchronizing the knee and ankle motion in hopping motions [64]. Based on this biomechanical insight, the effect of a GAS structure on the synchronization of knee and ankle is investigated in simulation and on the BioBiped2 robot. The effect of a GAS structure implemented as a linear spring is quantified by comparing the joint synchronization during a passive rebound experiment with and without the GAS structure attached.

7.2.1 Experimental Setup

The simulation model (see [61]) and the robot were set up to perform a passive rebound from a drop of fixed height, while the trunk is constrained to vertical translation (see Figure 7.1). Landing on its foot tips the robot was able to rebound only from the reaction of the passive springs in the elastic actuation. The legs were set up in different configurations, each describing the spring stiffness ratio of knee and ankle flexors as well as the initial angular difference

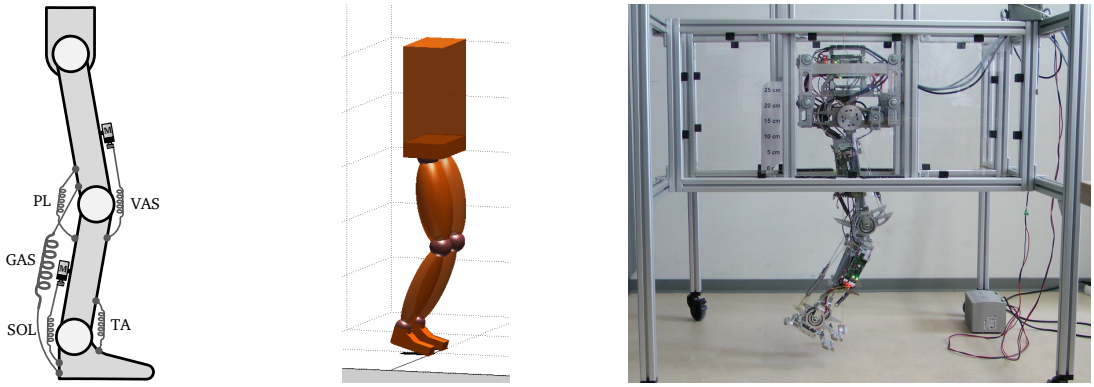


Figure 7.1: (a) Elastic structures used during the passive rebound experiment (b) Visualization of the BioBiped simulation model (c) BioBiped2 in the experimental setup for the robot experiments

Published in [70]

between knee and ankle joints. Absolute values for the knee and ankle joint angles were chosen so that the initial leg lengths were constant for all experiments. The initial hip angle was set to put the foot tips vertically below the hip joints. To gain insight into the effect of the use of a linear spring as GAS structure every experiment was performed once with and once without the structure attached. The rest length of the GAS structure was adjusted for each parameter configuration to correspond to the initial leg configuration.

7.2.2 Evaluation Criterion

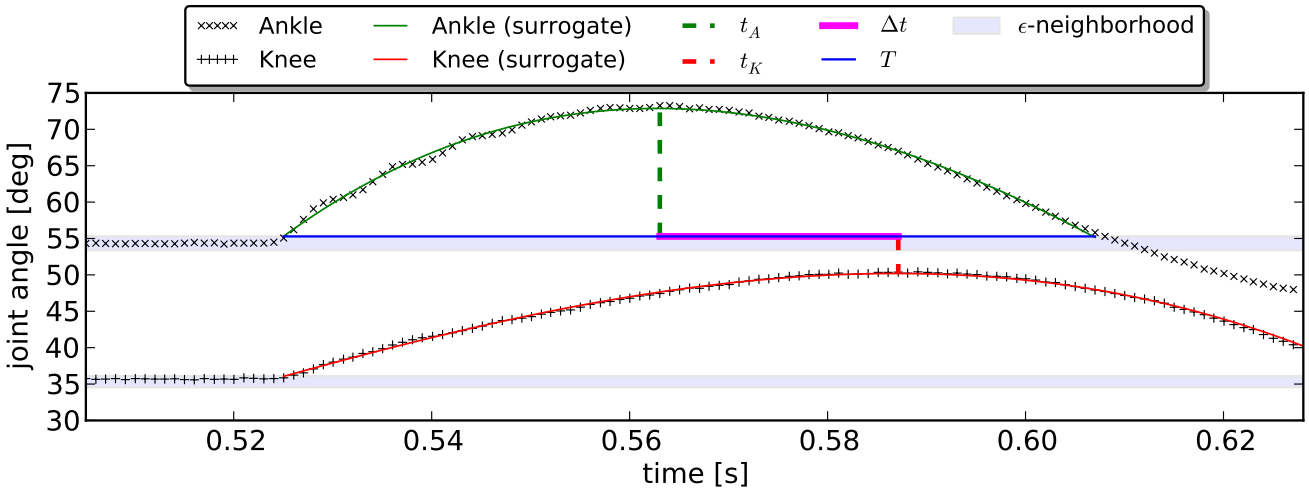


Figure 7.2: Visualization of the joint trajectories during one of the robot experiments. Besides the actual measurement data, the graph shows the surrogate functions, the total time T between leaving the ε-neighborhood and reentering it and the time difference Δt between the trajectories' maxima (t_A, t_K).

Published in [70]

To quantify the synchronization of knee and ankle motion a joint synchronization index is defined as phase difference $\Delta\phi$. This describes the ratio of the time difference Δt between the maximal joint deflection of the two joints and the total time T as can be seen in the following equation:

$$\Delta\phi = |\frac{t_K - t_A}{T}| = |\frac{\Delta t}{T}|$$

Where t_K and t_A are the times where knee and ankle are at their maximal deflection and T is defined as the time from the ground contact until one of the joints comes back to its original position (see Figure 7.2). To reduce the influences of measurement noise the calculation of t_K and t_A is made based on a surrogate function used to approximate the joint angle trajectories around the time of the maximal deflection. For a more robust detection of the total time T an ε-neighborhood is used around the initial joint angles with ε set to 5% of the maximal joint deflection. T is then calculated as the time from leaving the ε-neighborhood until the first joint reenters it.

7.2.3 Parameter Space

Two variable parameters are evaluated during the experiments:

spring stiffness ratio $R = k_{SOL}/k_{VAS}$ with k being the stiffness coefficient of the spring

initial joint angle difference $\Delta\theta = \theta_{K,0} - \theta_{A,0}$ with indices K and A referring to knee and ankle joint respectively

The other parameters are constant over all experiments. The initial leg angels are calculated from $\Delta\theta$ to result in a constant initial leg length L_0. L_0 is set to be 94% of the maximal leg length, which is the average value for humans at preferred hopping frequency (see [11]). The stiffness of VAS k_VAS was chosen to achieve a similar leg compression on rebound of about 10% of L_max. For SOL the stiffness k_SOL was then set according to the stiffness ratio R.

The values of the constant parameters are listed in the upper section of Table 7.1. In the two tables below are the values used for the variable parameters in the robot experiments. Each parameter combination was tested with and without the GAS structure attached. In the simulation experiment many more values in the same parameter range were used to gain more fine grained results.

7.2.4 Results

The synchronization index $\Delta\phi$ is shown in Figure 7.3 for the simulation experiments (top row) and the robot experiment (bottom row). The left and right column show the data plots without and with the GAS structure, respectively. Each plot displays the synchronization index $\Delta\phi$ over the initial angular difference $\Delta\theta$ (x-axis) and the spring stiffness ratio R (y-axis).

CONSTANT PARAMETERS								
L_{max}	[m]	0.727	k_{VAS}	[N/mm]	15.5	$F_{0_{\mathrm{VAS}}}$	[N]	36.8
L_0	[m]	0.94 L_{max}	k_{GAS}	[N/mm]	7.9	$F_{0_{\mathrm{GAS}}}$	[N]	27.6
			$k_{\mathrm{PL/TA}}$	[N/mm]	4.1	$F_{0_{\mathrm{PL/TA}}}$	[N]	13.8

STIFFNESS RATIO R IN ROBOT EXPERIMENTS						
EXPERIMENT		A	B	C	D	E
R	[-]	0.265	0.432	0.510	0.839	1.155
k_{SOL}	[N/mm]	4.1	6.7	7.9	13.0	17.9
$F_{0_{\mathrm{SOL}}}$	[N]	13.8	22.6	27.6	27.6	58.9

ANGLE DIFFERENCE $\Delta\theta$ IN ROBOT EXPERIMENTS						
EXPERIMENT		1	2	3	4	5
$\Delta\theta$	[deg]	-7	-0.5	6.6	14.8	24.7
KNEE $\theta_{K,0}$	[deg]	138	139.5	141.6	144.8	149.7
ANKLE $\theta_{A,0}$	[deg]	145	140	135	130	125

Table 7.1: Constant and variable parameters used during the experiments: leg lengths L_{max} and L_0, spring stiffnesses k and pretensions F_0, spring stiffness ratios R and joint angles θ.

Published in [70]

The results show that a synchronous operation of the joints is possible without the GAS, but only in a very narrow region (bright yellow to white in left column of Figure 7.3). To keep the robot operating in this region precise adjustments of the spring stiffness ratio and corresponding leg angle configurations would be needed to stay in the white region.

In the right column of Figure 7.3 it can be seen, that attaching the GAS structure broadens this region of synchronous joint operation. Thus allowing for a synchronous joint operation with fewer limitations on the values of R and $\Delta\theta$ and reducing the risk of unwanted heel strike, which dissipates energy.

7.2.5 Conclusion

The results in Figure 7.3 show that the implementation of a GAS structure as a passive linear spring improves the synchronization of the knee and ankle deflection in every tested parameter combination. Even though synchronous operation of the joints is also possible without the GAS structure in a narrow region of the parameter space, this region is significantly larger with the GAS attached. Therefore, implementing the GAS structure helps to avoid unwanted

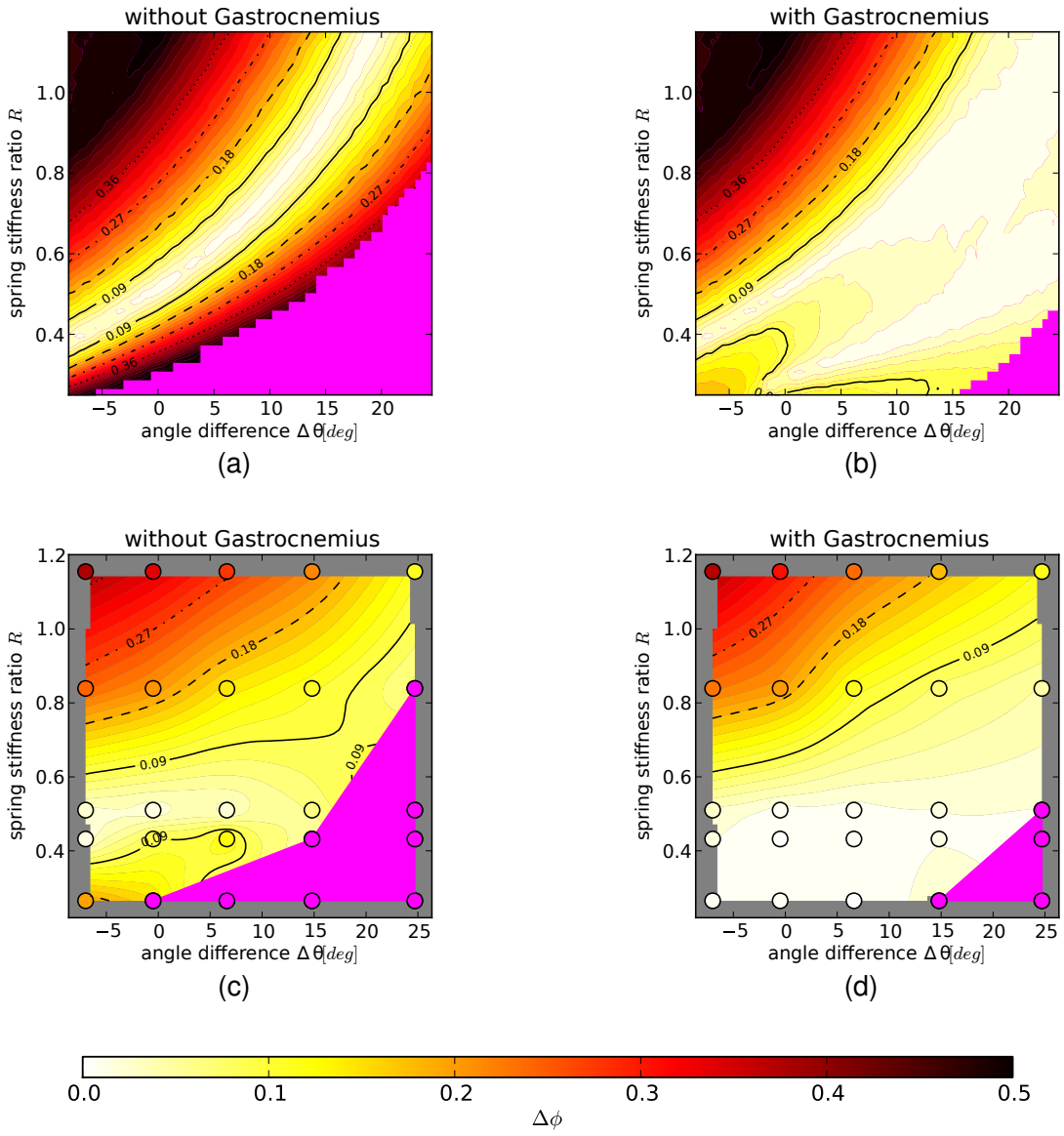


Figure 7.3: Phase differences $\Delta\phi$ of knee and ankle joints in the simulation (a, b) and in the robot experiments (c, d) each without and with GAS respectively. The trials where heel contact occurred during the stance phase are located in the lower right corner in both simulation and experiments and are marked in magenta. The configurations used for the 50 robot experiments (black circles) are shown in Table 7.1. As these configurations are not equidistant in the graph the $\Delta\phi$ values in-between the experiments have been linearly interpolated for easier comparison with the simulation results.

Published in [70]

heel contact and reduces the risk of overextension of knee and ankle joints for varied landing conditions. This improves the robustness of the segmented legs with respect to the time of the ground contact in the gait cycle which is especially important on uneven ground.

While these results show the advantages of implementing a GAS structure, they do not answer the question how to set up the GAS for a desired motion goal. Parameters of the GAS

include for example its rest length, the lever arm ratio and its spring stiffness. Testing all different configurations on the robot in a grid based pattern search as was done in this section is not feasible due to the high number of possible parameter combinations. From the results in shown in Figure 7.3 it can also be seen, that optimizing only in simulation would not suffice due to the differences to the robot results. Therefore, an approach is needed to efficiently perform a parameter optimization on the robot. A new concept to this is presented in the next section, combining simulation results with an expert guided hardware-in-the-loop motion optimization.

7.3 New Concept for Expert Guided Hardware-in-the-Loop Motion Optimization for Musculoskeletal Bipedal Robots

The verbatim contents of this section have been published in [71].

Conventionally built bipedal robots are based on rigid kinematic chains combined with stiff joint actuators which allow them to perform precise motions. Using established joint and posture control concepts they achieve stable fast walking motions and even short flight phases on flat ground [29]. But to allow for bipedal locomotion in unstructured environments as seen in humans, robustness against unforeseen disturbances with respect to the time and place of the ground contact is more important than precision. Leg kinematics with rigid actuators are subject to high peak forces on unplanned ground contacts which can lead to damages in the actuators. Also, the stiff nature of the kinematic chain does not allow to store and release energy between multiple steps to achieve human-like performance in running or jumping.

In biological legged systems such performance is achieved through elasticity in the actuation [74]. This can be transfered to mechanical systems by using series elastic actuators, which can passively reduce peak forces. Further, they can be used to store and release energy to make a cyclic gait more efficient by making the overall leg act as a spring. Implementing the actuation in a musculoskeletal arrangement, using tendon driven series elastic actuators (TD-SEAs) connected to the joints in an antagonistic setup, further advantages seen in humans can be exploited. The biarticular structures can help to synchronize the joint motions to avoid overextension of individual joints, increasing the robustness of the overall leg motion [70]. Also, their ability to transfer energy from proximal to distal joints can be utilized to design more lightweight extremities.

The BioBiped2 robot (see Figure 7.4 and [1]) which uses TD-SEAs for its joint extensors with passive springs as antagonists and biarticular structures is used in the example application in Section 7.3.2. This robot is the second generation robot based on the insights gained from the BioBiped1 platform described in [60]. It is also the first to which the approach proposed in this paper is being applied. Hardware improvements over the first generation include ball bearings in the joints, lower gear ratios for higher rope speeds and a modular electronics design to allow the use of more sensors and actuators depending on the current motion goal.

[1] `http://www.biobiped.de`

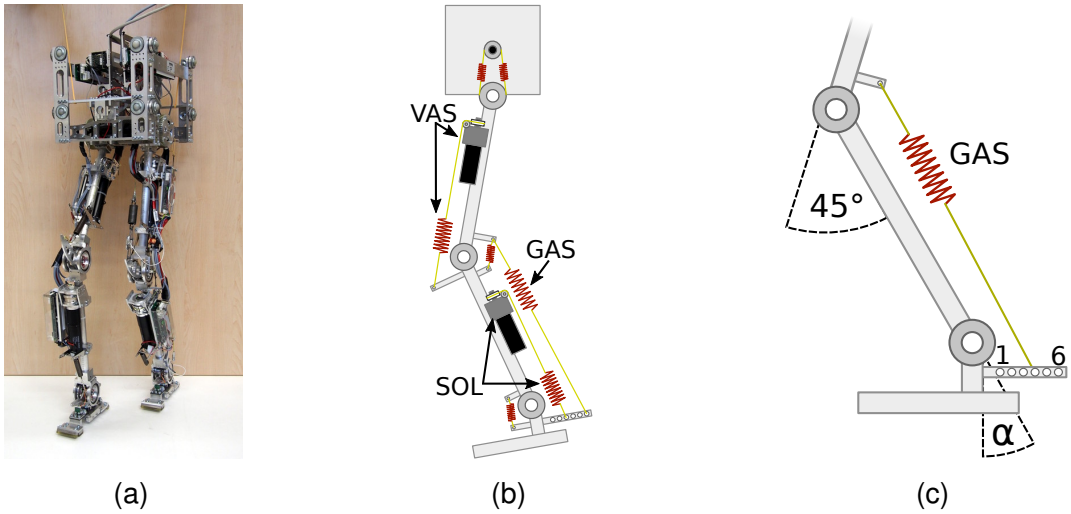


Figure 7.4: (a) BioBiped2 robot used in the example application in Section 7.3.2 (6 degrees of freedom, mass: 11.5 kg, hip height; 0.7 m). (b) Kinematic structure highlighting the relevant active TD-SEAs with motor and spring (Soleus (SOL), Vastus (VAS)) and passive (Gastrocnemius (GAS)) structures. (c) Definition of the GAS rest angle α and the GAS attachment point numeration 1 to 6.

To achieve a desired motion on such a musculoskeletal robot a number of parameters in hardware and software influencing the passive and active dynamics and control properties have to be designed and properly tuned. Using detailed multi-body system dynamics models like for conventional rigid robots is an even more difficult process for musculoskeletal robots. The highly elastic structures of the TD-SEAs, the dynamic interplay between multiple links and joints through biarticular structures and changing interaction with the ground during dynamic bouncing motions make a sufficiently accurate modeling of musculoskeletal robot dynamics highly difficult. Differences between simulation model and hardware that are non-trivial to remove include among others the non-linearities of physical springs, the unknown friction in joints and the modeling of the ground contact [34]. Therefore, optimization of these parameters only by robot dynamics simulation is not sufficient.

On the other hand using the actual robot in a hardware-in-the-loop optimization to find the best parameter values is very expensive with respect to time and can also be harmful to the robot hardware prototype. The number of robot experiments needed for the optimization depends on the number of parameters involved. In musculoskeletal robots the addition of elastic elements and biarticular structures increases this number compared to rigid robots. Each experiment has a wear on the robot's hardware, which restricts the number of experiments possible. Further, the time needed for manually modifying hardware parameters, like spring stiffness or lever arm length, makes each experiment costly. Damaging the hardware through use of unsuitable parameter combinations adds to the cost of the robot experiments in terms of repair time.

Therefore, a new approach is presented to reduce the costs of optimizing design and parameter tuning of musculoskeletal robots. The number of robot experiments needed is reduced through systematic interpretation of results of specifically designed simulation experiments. Furthermore, parameter combinations are excluded which are possibly harmful to

the hardware and by sequencing the experiments to have fewer hardware modifications in between them.

7.3.1 State of the Art

Design and tuning of elastic musculoskeletal bipedal robots has been described by Hosoda et al. [23, 38] and Niiyama et al. [47, 48].

In [23] the design, tuning and motion generation of a pneumatic biped which performs jumping, walking and running motions is described. The motion parameters are manually tuned for each of the motions performed without the use of simulation or optimization methods.

In [38] inertial measurement sensors in the trunk of the robot are used to detect its roll angle for stabilization of a rebound motion. Since the experiments are performed only on the hardware without help of simulation models, hundreds of trials on the robot are needed to collected sufficient data.

While in [48] a simulation model is used to adapt human muscle activation patterns to the Athlete robot simulation model, the resulting parameters are still manually tuned on the robot afterwards.

In [60] the BioBiped1 musculoskeletal robot performs synchronous and alternate hopping motions using a manually tuned parameters.

All mentioned approaches, as well as the experiments carried out on BioBiped versions 1 and 2 so far, used robot parameters manually tuned directly on the hardware without systematic exploitation of simulation results.

7.3.2 Expert Guided Optimization by Example

The goal of this work is to efficiently determine a parameter configuration for possibly optimal motion of the robot while keeping the number of hardware experiments and the time consumed by performing them low. To find the optimal values for the relevant motion parameters a hardware-in-the-loop optimization is performed which is guided by a human expert. This expert reduces the number of hardware experiments by applying knowledge about the robot's behavior gained from previous experiments, biomechanical understanding of the system, and interpretation of results from simulation experiments. As the knowledge from previous experiments and the biomechanical understanding are difficult to exploit in a systematic and reproducible manner, this work will focus on the systematic generation, interpretation and usage of the simulation results. This approach is split into the four steps shown in Figure 7.5 which are detailed in the following sub-sections.

The approach is applied to the musculoskeletal bipedal robot BioBiped2 shown in Figure 7.4, which uses TD-SEAs based on DC-motors, synthetic ropes and metal extension springs as actuators. For the simulation experiments a multibody system (MBS) simulation model is used, that was developed for the BioBiped robot series in [61].

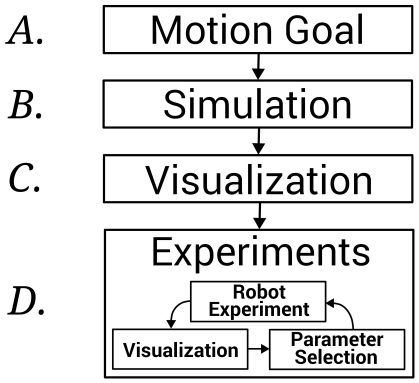


Figure 7.5: Overview of the steps performed in the optimization process.

The overall goal of the BioBiped project is to perform different gaits on a single robot configuration from jogging to walking to stable standing. As first step towards jogging with this new robot model, hopping is considered. The performance of a synchronous hopping motion, including impacts and push-offs, is optimized here as a prerequisite for future jogging motions. While in this robot multiple bi-articular structures can be attached, in this example only the bi-articular GAS is used because of its relevance for the considered hopping motion.

A. Definition of Motion Goal and Optimization Settings

The motion goal needs to be defined including a quality criterion which can be measured or derived for both the simulation and the hardware experiments. Using the human leg as model, the biomechanical understanding of its functional structures is used to identify which of the robot's structures are relevant for the selected motion goal.

The goal of the example optimization is to improve the hopping performance in a synchronous hopping motion. From biomechanics it is known, that human hopping is primarily powered by ankle motion [11]. Therefore, the mechanical structures in the BioBiped2 robot most relevant for this motion are the active mono-articular ankle extensor SOL and the passive bi-articular GAS.

So the parameters $\boldsymbol{p}$ subject to the optimization performed in this example application are the stiffness of the SOL and GAS structures as well as the rest length and lever arm of the GAS. For the SOL and GAS stiffness five different springs are available with their parameters listed in Table 7.2. The GAS structure has a fixed lever arm length on the thigh and six possible attachment points at the heel (shown in Figure 7.4(c)) with their distance from the center of the joint listed in Table 7.2. Its rest length is the only continuous parameter which is described through the knee and ankle angles corresponding to its rest position. This is the most practically viable approach on the robot, since both joints feature position encoders, which can be used to measure the currently set rest length. Positioning the knee joint at $45\,\mathrm{deg}$ bent from full extension the adjustable GAS rest length corresponds to ankle joint angles between $0\,\mathrm{deg}$ and $40\,\mathrm{deg}$ bent from center position. This GAS rest angle α is defined as shown in Figure 7.4(c) (with the SOL disengaged).

The objective of the optimization is to minimize the quality value $q \in \mathbb{R}$. It depends on the vector of design parameters $\boldsymbol{p}$, which may include real- and integer-valued parameters and which are to be tuned by the expert guided optimization approach. The quality q of the hopping is calculated from the duty factor q_{df} and the maximal center of mass (CoM) height q_{com} as shown in Equation (7.1). The CoM position is located in the lower trunk in straight standing, which is used as the fixed reference point for the hopping height measurements

$$q(\boldsymbol{p}) = \frac{\hat{q}_{df}(\boldsymbol{p}) + \hat{q}_{com}(\boldsymbol{p})}{2} \,. \tag{7.1}$$

Using just one of them for the quality might allow for non-hopping motions to achieve good quality values, e.g. by just pulling up the feet for a low duty factor or just standing on fully extended legs for a high CoM height. The combination of both ensures an actual hopping motion with flight phase and upward motion of the CoM. To ensure an equal weight of both parts they are normalized based on the minimal and maximal values found in the simulation coverage experiments: $\hat{q}_{df}(\boldsymbol{p}) = (q_{df}(\boldsymbol{p}) - q_{df}^{min})/(q_{df}^{max} - q_{df}^{min})$, $\hat{q}_{com}(\boldsymbol{p}) = (q_{com}(\boldsymbol{p}) - q_{com}^{min})/(q_{com}^{max} - q_{com}^{min})$. Also, to formulate this as a minimization problem q_{com} is set to the negative maximal CoM height of one motion cycle. The values of the two parts are calculated as shown in Equation (7.2):

$$q_{df}(\boldsymbol{p}) = \frac{t_{stance}}{t_{stance} + t_{flight}}, \; q_{com}(\boldsymbol{p}) = -h_{max} \tag{7.2}$$

In simulation the maximal CoM height h_{max} can be directly read from the model as the highest point of the CoM trajectory during flight phase. For the robot experiments this value is calculated as a combination of accelerometer and kinematic data. The vertical position of the trunk is calculated from the measured joint angles via forward kinematics during ground contact and the accelerometer data is used to calculate the trajectory during the flight phase. The drift of the accelerometer is compensated using the heights of the trunk known from the kinematics just before and after the flight phase.

The ground contact forces are used to divide the motion into stance and flight phase for both the simulation and the robot. The duty factor q_{df} is calculated as stance time t_{stance} in relation to the time of a hopping cycle $t_{stance} + t_{flight}$. A minimal vertical force value of $10\,N$ is used to detect ground contact for the simulation and the robot to ensure equal calculations for both.

As safety criterion for the robot the maximal forces f_{max}, that occur at the actuation structures for SOL, GAS and VAS, are compared to the force limit f_{limit}^{spring} of the spring currently used in the respective structure on the robot which can be seen in Table 7.2. Configurations where the limits of any of the three springs are exceeded as shown in Equation 7.3 are marked in the visualization and excluded from the robot experiments to protect the mechanics.

$$f_{max}^{x} > f_{limit}^{spring}, \quad x \in [SOL, GAS, VAS] \tag{7.3}$$

While hopping, the robot is stabilized by an external mechanism constraining its trunk motion to vertical translation. The hopping is performed on flat ground and the motor power supply is limited to maximal output of $10\,V$ to protect the system.

Attachment point	1	2	3	4	5	6
Distance [mm]	45.3	51.1	56.8	62.7	68.5	74.5

Spring constant [N/m]	4100	7900	10000	13000	15600
Force limit [N]	162.8	341.5	356.7	341.5	386.7

Table 7.2: Parameter values of the available attachment points and springs.

The motion trajectory is generated by a state machine with two states switching between a bent and an extended leg configuration. Transitions between the two states are triggered by the ground contact events touchdown and liftoff and trajectory transitions are smoothened by a spline interpolation from the current actuator positions to the new goal positions. The tracking of the trajectories is performed by a motor position controller with the same manually tuned gains in simulation and on the robot.

Initially the robot is in the bent configuration and is dropped manually from a height with 5 cm ground clearance.

B. Design of Simulation Experiments

The simulation experiments are designed to achieve three goals:

- Understanding the sensitivity of the quality criterion,
- Recognizing correlations of multiple parameters,
- Selecting a starting point for the robot experiments.

For the first two goals a coverage of the parameter space is needed and for the third an optimization in simulation is used to find a good starting point.

To achieve a good coverage of the parameter space with the simulation in feasible time the continuous parameter GAS rest angle is discretized into nine values. Together with the three discrete parameters this results in a total number of parameter configurations for the coverage simulation experiments of $9 * 5 * 5 * 6 = 1350$. With an average of 10 s needed to simulate the experiment for one configuration the approximate total time needed for simulation is $3h\,45m$, which allows for a full factorial design of experiments [4].

To optimize the continuous parameter this nonlinear function with continuous and discrete variables a mixed-integer nonlinear problem (MINLP) has to be solved without gradient information. A surrogate based mixed-integer nonlinear black box optimization is chosen [18], which can make use of the already extensive data gained in the coverage experiments as initial data set for its surrogate function.

C. Visualization and Interpretation of Simulation Results

The goal in this step is to systematically leverage the results from the simulation experiments to help plan the robot experiments to be as efficient as possible. Mapping the simulation

results to the robot results is difficult to automate, since the model error of the simulation and any inaccuracies in setup of the hardware are not known. Results from the robot experiments could be used to improve the simulation accuracy, but this is beyond the scope of this work. Therefore, a systematic approach is used to leverage the knowledge gained by interpreting the simulation results with the help of visualization of the quality criterion.

As the parameter space has four dimensions plus the dimension of the quality criterion the visualization has to be split into multiple plots. A two dimensional grid of two dimensional plots was chosen with the quality criterion represented through color as can be seen in the overview Figure 7.6. Due to space constraints only a subset of the parameter space is visualized in more detail in this publication. For the reduction of the parameter space three plots showing different sectional planes through the optimal configuration found in simulation are shown in Figure 7.7.

Exclusion of harmful parameter configurations

As can be seen in the overview Figure 7.6 and in the detailed plots in Figure 7.7 only a few harmful configurations, marked as magenta diamonds, were identified in simulation based on Equation (7.2). These configurations lead to maximal forces in one of the three elastic structures of SOL, GAS or VAS that were higher than the specified force limit of the springs to be used on the robot. To protect the robot from damage, these configurations will be excluded from the robot experiments.

Adjustment of the quality criteria visualization boundaries

The upper boundary for the quality criterion is set to 0.7 in the visualizations shown in this paper. This value was manually selected by the expert to focus on the relevant area of the parameter space and clearly show the differences in the quality around the optimal value as can be seen in Figure 7.7.

Exclusion of parameters

In Figure 7.7(b) it can be seen that the stiffness value of the GAS structure has only a very small influence on the quality criterion, but cannot completely be excluded from the optimization.

Recognition of parameter correlations

By visualizing all sectional planes of the parameter space as shown in Figure 7.7, linear correlations between all parameter combinations can be visually inspected. In this example application, a linear correlation is only found between GAS rest angle and GAS attachment point as shown in Figure 7.7(c). This information is used in the next section when planning the robot experiments.

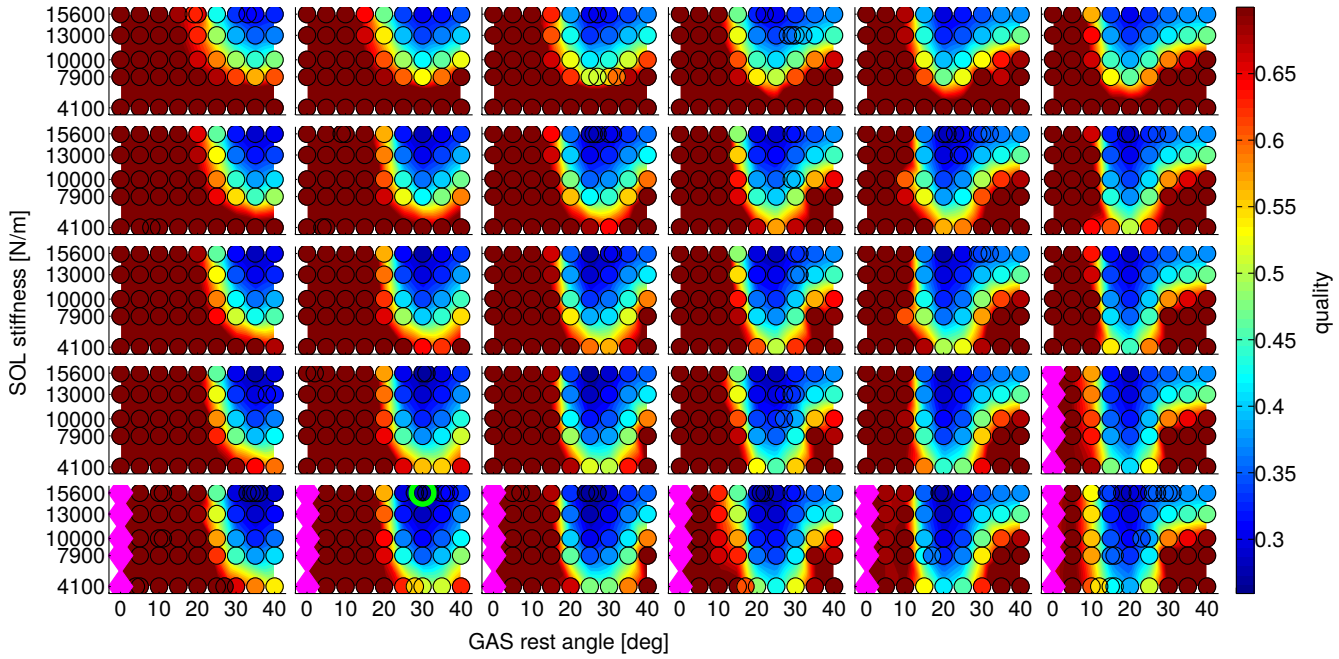
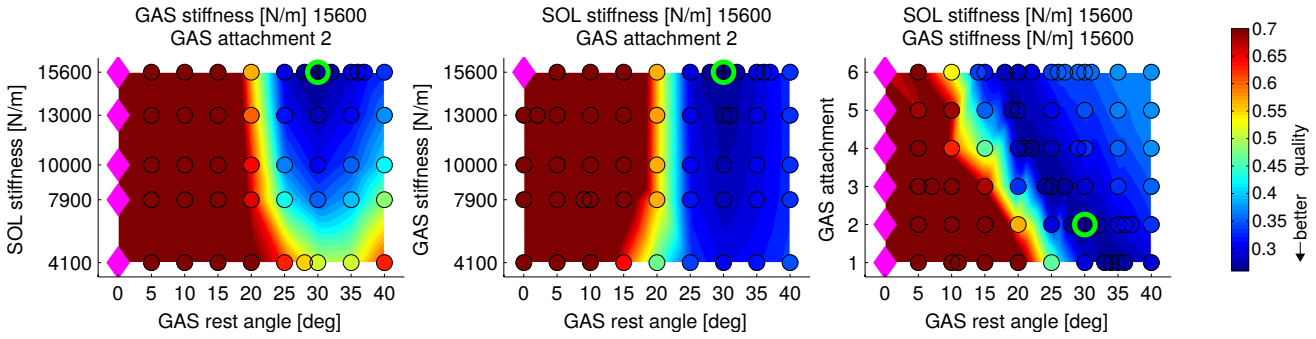


Figure 7.6: An overview of the results gained through the simulation coverage experiments spread over the parameter space. All plots show the same sectional plane of the parameter space with the GAS rest angle on the x-axis and the SOL stiffness on the y-axis. The plots in each column share the same GAS attachment point 1 to 6 from left to right. The plots in each row have the GAS stiffness in common, with the lowest spring coefficient in the top and the highest in the bottom row. A more detailed excerpt of the plot with the best value can be found in Figure 7.7(a). (black circles: simulation experiments colored with quality values (best values are dark blue), magenta diamonds: harmful configurations, green circle in bottom row second column: the best value found)



(a) The strongest influence on the quality criterion is the rest angle of the GAS structure, but also the stiffness of the SOL is relevant.

(b) The stiffness of the GAS structure has very small influence on the quality criterion in the area with good results colored in blue.

(c) A correlation can be seen between the rest angle and the attachment point of the GAS structure.

Figure 7.7: Three different sectional planes of the simulation results cut through the best configuration. (black circles: simulation experiments colored with quality values, magenta diamonds: harmful configurations, green circle: the best value found)

Based on the interpretation of the simulation results the robot experiments can now be planned and executed in a more efficient manner.

Design for the initial robot experiments

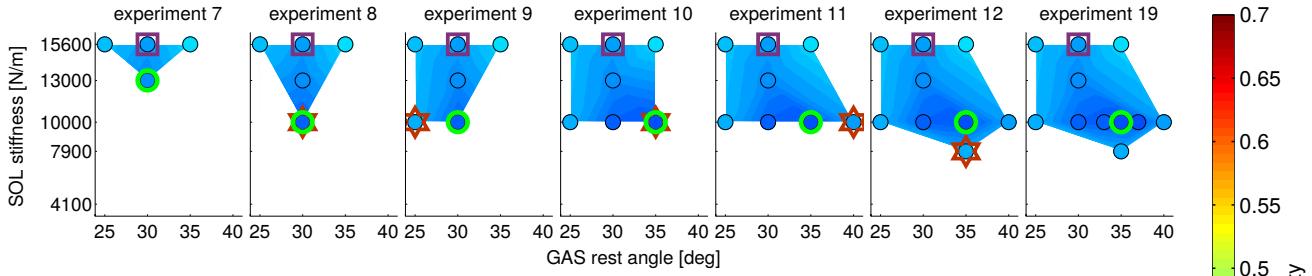
First a start configuration for the robot experiments has to be selected. Based on the simulation results, it is safe to use the optimal configuration found in simulation, as no harmful configurations are close to it. The initial robot experiments are planned around the start configuration varying each parameter by a single step in both directions as proposed in [4] with the central finite differencing approach described in [4]. The step size is chosen for the discrete parameters to be one step and for the continuous GAS rest angle to be the size of its discretization. As these step sizes show significant changes in the quality criterion, this will give the expert a first impression of the gradients of each parameter on the robot and allow for a visual mapping between simulation and robot results.

Execution and further selection of robot experiments

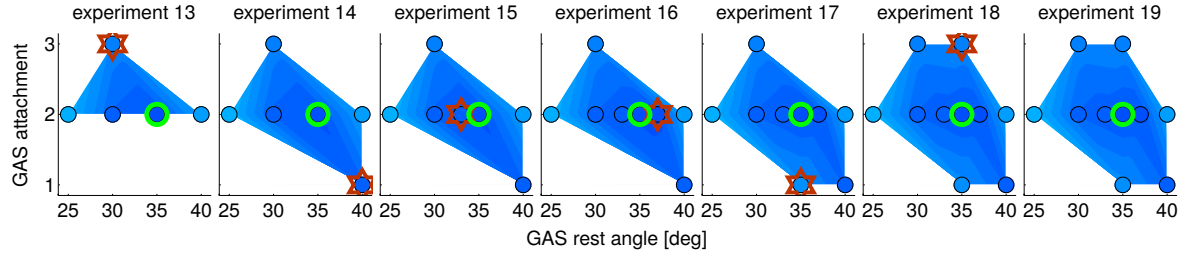
After the seven initial experiments, the results are visualized, shown in the top left plot in Figure 7.8(a) entitled 'experiment 7'. Here it can be seen, that the best configuration so far (marked with a green circle) has a lower SOL stiffness than the optimum found in simulation (marked with a purple square). Following the gradient in the quality value towards the next lower SOL stiffness value reveals an even better result in 'experiment 8'. By following this gradient further along the SOL stiffness and GAS rest angle parameters a local optimum is found in 'experiment 10'. As the visualization after 'experiment 12' shows that the SOL stiffness parameter set to $10000\,\frac{N}{m}$ leads to the best results in this sectional plane, the search is continued in the sectional plane between GAS attachment point and GAS rest length shown in Figure 7.8(b). Due to the linear diagonal correlation found in this plane in the simulation experiments, neighboring configurations along this correlation are tested in experiments 13 and 14 as shown in Figure 7.8(b). In experiments 15 and 16 more fine grained changes of $2\,\mathrm{deg}$ to the continuous GAS rest angle parameter are tested with no further improvement of the quality criterion. With experiments 17 to 19 all direct neighbors of the best configuration found so far are tested.

Termination of the robot experiments

The termination criterion used in this example application is the confirmation of a local optimum. To ensure a local optimum also the neighboring values along the parameter correlation found in simulation between GAS rest angle and GAS attachment point have been tested.



(a) The graphs show the plane with GAS stiffness fixed at $15600\frac{N}{m}$ and GAS attachment in position 2. After the seven initial robot experiments around the start configuration, the gradient is followed in experiments 8 to 12 to find the best values for the stiffness of SOL and the rest angle of GAS.



(b) The graphs show the plane with GAS stiffness fixed at $15600\frac{N}{m}$ and SOL stiffness fixed at $10000\frac{N}{m}$. In experiments 13 and 14 the attachment point and rest length of GAS were optimized by first testing the two points along the diagonal correlation of the two parameters known from the simulation results (Figure 7.7(c)). Then in experiments 15 and 16 the continuous rest angle of GAS was modified more fine grained than in the discretized steps used before. Finally, in experiments 17 to 19 the best configuration found so far was confirmed to be at least a local optimum.

Figure 7.8: Iterative construction of the visualization for the results of the robot experiments. Markers show the best configuration from simulation as purple square, the best configuration on robot so far as green circle and the newly added result as red star.

7.3.3 Comparison to Surrogate Based Optimization Method

To be able to evaluate the proposed approach, a conventional hardware-in-the-loop optimization is applied to the example application for comparison. The optimization problem is formulated as a minimization problem using the quality criterion q as described in Equation (7.1). Additionally, the safety criterion described in Equation (7.3) is used based on the simulation data to identify harmful configurations before they are tested on the robot. When the optimization chooses to evaluate such a harmful configuration, it is not performed on the robot, but marked as infeasible for the optimization. The optimization terminates, when the quality criterion converges or the number of robot experiments after the initial design is twice that of the expert guided approach, namely at experiment 31.

The optimization has to be performed with a problem solver capable of handling mixed-integer nonlinear problems. A surrogate based mixed-integer black box optimization approach (SurOpt) [18] has been shown to find good parameter configurations with a low num-

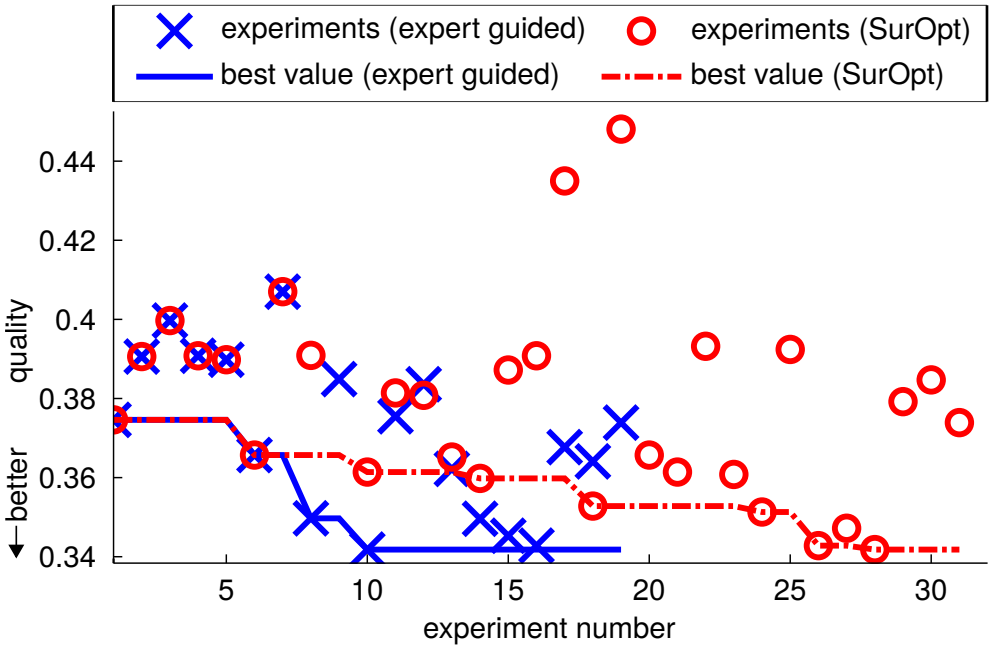


Figure 7.9: Comparison of the quality criterion value over the course of the expert guided and SurOpt optimization robot experiments. It can be seen that the same seven initial configurations have been used on both approaches. Afterwards the expert guided approach improves the quality criterion much quicker and finds the best solution in experiment 10 while SurOpt finds the same configuration only in experiment 28.

ber of hardware experiments for related problems [19]. Therefore, it is a valid candidate for comparison with the expert guided approach presented in this work.

The parameters to be optimized are the same as in the expert guided approach: SOL stiffness, GAS stiffness, GAS attachment point and GAS rest angle. The ranges of these parameters are normalized to be mapped to the ranges from 0.0 to 1.0 to allow for an efficient search in all parameter dimensions. A branch and bound approach is used to handle the three discrete parameters. A distance based update criterion is used in this optimization (compare [18], Chapter 4.2), which enforces a minimal distance ε between tested configurations. The value of ε is chosen to be 0.025, corresponding to changes in the GAS rest angle of 1 deg, which is the minimal change that is practically feasible on the robot.

As suggested in [18], expert's guesses are used for the initial design points. Here the optimization is started on the robot with the same initial design around the simulation optimum. Sequential updates to the parameter configuration are selected by the optimization algorithm to either improve the quality criterion or the surrogate function mean square error as described in [19]. The configurations are tested on the robot and evaluated in the same manner as for the expert guided experiments making the resulting quality values directly comparable.

The optimization found the same solution as the expert guided approach after 28 robot experiments, but its termination criterion of converging results was not yet fulfilled. After 31 experiments the optimization was stopped with the maximum number of experiments defined as second termination criterion.

	SOL	GAS			results			num.
	stiffness [N/m]	stiffness [N/m]	attach-ment point	rest angle [deg]	q	q_{com} [m]	q_{df}	robot experi-ments
manual tuning	15600	7900	3	30	0.443	0.783	0.383	14
simulation	15600	15600	2	30	0.375	0.795	0.352	0
expert guided	**10000**	**15600**	**2**	**35**	**0.342**	**0.805**	**0.354**	**19**
SurOpt	10000	15600	2	35	0.342	0.805	0.354	31

Table 7.3: Shown are the parameter configurations $\boldsymbol{p}$ for the best (minimal) quality values $q(\boldsymbol{p})$ found with manual tuning, the simulation optimization, the expert guided approach and the SurOpt optimization. The resulting quality values were produced on the BioBiped2 robot. The number of robot experiments needed to find these configurations are listed in the last column.

The best parameter configurations $\boldsymbol{p}$ found by manual tuning, the simulation optimization, the expert guided approach and the SurOpt optimization with their corresponding quality values $q(\boldsymbol{p})$ and maximal CoM heights evaluated on the BioBiped2 robot are listed in Table 7.3. The quality from the manually tuned result of 0.443 is improved already by using the optimal configuration found in simulation on the BioBiped2 robot which leads to a quality value of 0.375. Further improvement was possible using the expert guided approach and the SurOpt hardware-in-the-loop optimization, both resulting in the same optimal configuration with a quality of 0.342.

While the expert guided approach and the SurOpt optimization both find the same optimal configuration the former needs fewer robot experiments as shown in Figure 7.9. It can be seen that it took SurOpt 28 experiments to find this optimum while the expert found it on the 10th experiment.

To better understand the results the quality value can be split into its two parts, the maximal center of mass height q_{com} and the duty factor q_{df}. While the quality improvement through the simulation stems from both the maximal CoM height and the duty factor, the improvement made in the robot experiments comes from an increased maximal CoM height. Compared to the manually found configuration the optimal configuration found in simulation already results in an improvement of the maximal CoM height of $12\ mm$. But the optimal solution found by both the expert guided and the SurOpt optimization gives an even higher gain of $22\ mm$ while the duty factor is almost the same as found through simulation.

7.3.5 Conclusion

This work introduces a systematic approach to optimize parameters of a musculoskeletal bipedal robot efficiently by reducing the number of needed hardware experiments through exploitation of simulation results. By systematic interpretation of the simulation results an expert can plan the hardware experiments to be more efficient than a state-of-the-art hardware-in-the-loop optimization method.

A parameter optimization of the musculoskeletal BioBiped2 robot to increase hopping performance was used as an example application to compare this new approach with a state-of-the-art hardware-in-the-loop optimization method. The parameters selected for optimization all had significant influence on the quality criterion, except for the stiffness of the GAS. As the quality criterion is a performance criterion which reflects the hopping height of the musculoskeletal robot, this can be explained through the biomechanical understanding of the role of the GAS structure. Its main purpose is to distribute power between the knee and ankle joints and not to store and release energy in its elastic element. Therefore, its elastic property is not as important for the quality criterion when compared to its other two parameters, rest length and lever arm, which shape the kinematics of the power transfer. The other two parameters of the GAS (lever arm length and rest length) showed significant influence on the hopping performance. As the role of the GAS in human locomotion includes powering the push-off of the leg before the swing phase, it can be concluded that optimizing its parameters is important to improve the locomotion performance.

In this example application the newly presented expert guided approach needed a total number of 19 hardware experiments to find and validate the optimal configuration. While the state-of-the-art optimization method found the same solution, 31 experiments were needed and no validation of it to be at least a local optimum was included. Further, through the expert guided sequencing of the experiments less time was needed for the hardware modification between experiments. In total the newly presented approach needed only 61% of the robot experiments and 52% of the time for the experiments and hardware modifications compared to the other optimization method while finding the same result.

Although the presented approach can be applied to general robot designs as well, it is expected to be most beneficial for highly complex robot designs such as musculoskeletal robots. For such robots with biomechanically inspired elasticity and damping properties, optimally balancing passive and active dynamics and control properties through parameter tuning is less effective with existing approaches.

8 Conclusion

Currently, most bipedal robots that can perform stable walking are designed and built as kinematic chains of rigid joints and links. Even though it was demonstrated that they can perform stable walking and also slow jogging with small flight phases their performance and efficiency is quite limited compared to human locomotion.

This is in part due to the lack of elasticity and compliance, which is an essential part of the human tendon driven actuation system. However, this significantly complicates design and control if purposely introduced to a robot's joint actuators. Using a musculoskeletal leg design with tendon driven series elastic actuators as investigated in this thesis for the BioBiped robot series allows exploiting the compliant properties to protect actuators from impacts and the elastic properties to conserve energy between steps as shown in Section 6.1.1. Further the biarticular structures in a musculoskeletal setup allow for power transfer towards distal joints and help synchronize multiple joints preventing overextension as shown in Section 7.2.

But by adding elasticity and biarticular couplings the complexity with respect to setup, control and modeling of such musculoskeletal robots is strongly increased. The elasticity and the additional couplings raise the requirements on the control system (see Chapter 3) and the control concepts (see Chapter 5) needed to perform motions of high quality. An new approach for the control of such a complex robot with highly nonlinear motion dynamics based on a learned inverse dynamics model has been investigated successfully in Section 5.3. The results show the validity of this approach for this problem, but the effort needed for the model generation and its adaptation to hardware changes is relatively high.

To enable the use of a large variety of novel control methodologies as such model based, multi-variable control approaches, a specific electronic control system architecture has been developed and successfully been applied and validated in this thesis for the BioBiped robot series (Section 4.4). It is a modular architecture to allow for extensibility of the robots with additional sensors and actuators over the evolution of the prototype generations. But still allows for true multi-variable control, without the use of local controllers at each actuator, which is used in most other bipedal robots. This is achieved by providing a central control system with all low-level sensor data over a low latency, high bandwidth EtherCAT bus connection. With enough computational power, this central control system can be used for online, model based control approaches under real-time constraints.

For the development of a new prototype series of robots it is important to improve the mechanical and electronic design from one robot generation to the next. To identify possible areas of improvement for the hardware, systematic evaluation of basic functionalities of the robot are needed as described in Chapter 6. With these experiments the hardware design of the BioBiped robot series has been evaluated and advanced successfully over three generations as detailed in Section 4.6.

During development and operation of robotic systems suitable tools are needed to to monitor the robot's behavior and and (re-)configure its motion parameters to progress efficiently. Since no integrated graphical user interface (GUI) existed for the used robotic middleware ROS, a new GUI framework was developed in this thesis originally for application to the Bio-Biped robots. The code was also released as open source in the rqt project which has become the standard GUI for ROS. It is now being used worldwide in robotics research and development as detailed in Section 4.5.2.

In musculoskeletal robots, the increased number of parameters involved in the actuation of each joint due to the additional elastic elements and joint couplings makes the setup of such a robot more difficult. The use of well suited parameter values plays an important part in the exploitation of the potential advantages of the highly elastic musculoskeletal robot. Due to the difficulties in accurately modeling and simulating such a complex robot there is a non-negligible difference between simulation and robot (see, for example, Figure 7.3). Consequently, an optimization of the parameters only in simulation is not sufficient, rather the real robot has to be involved in the optimization process. However, the costs of hardware experiments are a combination of time needed for hardware setup, modifications and experiments as well as wear and even damages to the robot that require repair time. To keep the overall costs of parameter optimization low, it is important to reduce the number of hardware experiments.

Therefore, the new concept presented in this thesis in Section 7.3 allows reducing the number of hardware experiments needed for parameter optimization. A systematic approach is described and validated that uses expert guided robot experiments to efficiently find good parameter values on the robot. The expert uses biomechanical understanding of the functional structures of the human leg and the visual interpretation of specific simulation results to plan and conduct the robot experiments efficiently. Further the tailored use of simulation results helps to reduce the risk of damaging the robot due to harmful parameter values.

This new concept has been validated in an example application for the musculoskeletal BioBiped2 robot. As a prerequisite for the investigation of jogging motions, the performance of a synchronous two-legged hopping motion has been optimized. The hardware parameters of the biarticular Gastrocnemius and mono-articular Soleus structures at the ankle joint are the most relevant for hopping in humans. Simulation experiments were used to identify correlations between the parameters and find a suitable start configuration for the robot experiments. With the expert guided robot experiments it was possible to improve the hopping performance compared to the previously used manually tuned parameter values and also compared to the optimal values found in simulation.

As benchmark for its efficiency, this new concept was compared to a state-of-the-art hardware-in-the-loop optimization solving the same problem. While basically the same parameter values were found by both approaches the expert guided approach needed 39% fewer robot experiments to find the solution and validate it at least as a local optimum.

The new concept presented in this thesis offers a systematic approach to efficiently optimize parameters directly for the musculoskeletal robot. It yields better results than manual

tuning of the robot's parameters or their optimization in simulation. Compared to a state-of-the-art hardware-in-the-loop optimization it significantly lowers the costs of the optimization process by reducing the number of robot experiments needed. This makes it a greatly useful methodology for improving the performance of highly elastic musculoskeletal robots.

Acknowledgements

This thesis was created during my time as a research assistant at the *Simulation, Systems Optimization, and Robotics Group (SIM)* at the *Department of Computer Science* of the *Technische Universität Darmstadt.* Parts of my research have been supported by the *German Research Foundation (DFG)*.

I would like to thank my supervisor Prof. Dr. Oskar von Stryk for his patience and support over the last years. His dedication to robotics research gave me the opportunity to work with a lot of robots and do research in this very interesting and challenging field. I am very thankful for all the encouraging and fruitful discussions.

Likewise I want to thank my second referee Prof. Dr. André Seyfarth for his interest in my work and his valuable and reassuring feedback.

I thank all my former colleagues for making my time in the group absolutely enjoyable and being very supportive. Special thanks go to Stefan Kurowski for all the helpful discussions and also the sometimes very much needed distractions. My acknowledgement also goes to Katayon Radkhah for allowing me to use her multibody simulation model.

Most of all I want to thank my family and friends who supported me during my studies. Especially Frieda for all her patience, but also her encouragement to finish this thesis. Also I am most grateful to Jan for sacrificing his weekends to read each chapter and motivating me to move forward.

Bibliography

[1] R. M. Alexander, „Energy-saving mechanisms in walking and running,“ *Journal of Experimental Biology*, vol. 160, no. 1, pp. 55–69, 1991. [Online]. Available: http://jeb.biologists.org/content/160/1/55.short.

[2] P. Allgeuer, M. Schwarz, J. Pastrana, S. Schueller, M. Missura, and S. Behnke, „A ROS-based software framework for the NimbRo-OP humanoid open platform,“ in *Proceedings of 8th Workshop on Humanoid Soccer Robots, IEEE Int. Conf. on Humanoid Robots, Atlanta, USA*, 2013.

[3] A. A. Biewener and T. J. Roberts, „Muscle and tendon contributions to force, work, and elastic energy savings: a comparative perspective,“ *Exercise and sport sciences reviews*, vol. 28, no. 3, pp. 99–107, 2000.

[4] R. Brekelmans, L. Driessen, H. Hamers, and D. Hertog, „Gradient estimation schemes for noisy functions,“ English, *Journal of Optimization Theory and Applications*, vol. 126, no. 3, pp. 529–551, 2005, ISSN: 0022-3239. DOI: 10.1007/s10957-005-5496-2. [Online]. Available: http://dx.doi.org/10.1007/s10957-005-5496-2.

[5] W. Chung, L. Fu, and S. Hsu, „Motion control,“ in *Springer Handbook of Robotics*, 2008, ch. 6, pp. 133–159. DOI: 10.1007/978-3-540-30301-5_7. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30301-5_7.

[6] M. R. Dawson, C. Sherstan, J. P. Carey, J. S. Hebert, and P. M. Pilarski, „Development of the Bento arm: an improved robotic arm for myoelectric training and research,“ in *MEC '14: Myoelectric Controls Symposium*, 2014.

[7] J. Dean and A. Kuo, „Elastic coupling of limb joints enables faster bipedal walking,“ *Journal of the Royal Society Interface*, vol. 6, no. 35, pp. 561–573, 2009. [Online]. Available: http://rsif.royalsocietypublishing.org/content/6/35/561.short.

[8] R. Dumas, L. Chèze, and J.-P. Verriest, „Adjustments to McConville et al. and Young et al. body segment inertial parameters,“ *Journal of Biomechanics*, vol. 40, no. 3, pp. 543–553, 2007, ISSN: 0021-9290. DOI: http://dx.doi.org/10.1016/j.jbiomech.2006.02.013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021929006000728.

[9] R. Edlinger, M. Zauner, and W. Rokitansky, „RRTLAN - a real-time robot communication protocol stack with multi threading option,“ in *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, 2013, pp. 1–5. DOI: 10.1109/SSRR.2013.6719319.

[10] *EtherCAT - the Ethernet Fieldbus*. [Online]. Available: http://www.ethercat.org/en/technology.html (visited on Jul. 21, 2015).

[11] C. T. Farley and D. C. Morgenroth, „Leg stiffness primarily depends on ankle stiffness during human hopping,“ *Journal of Biomechanics*, vol. 32, no. 3, pp. 267–273, 1999.

[12] J. Geisler, „Derivation and modelling of the equation of motion of a two-legged musculoskeletal walking robot,“ Master's thesis, TU Darmstadt, 2014.

[13] M. Gienger, K Loffler, and F. Pfeiffer, „Towards the design of a biped jogging robot,“ in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, IEEE, vol. 4, 2001, pp. 4140–4145. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=933265.

[14] J. Grizzle, J. Hurst, B. Morris, H.-W. Park, and K. Sreenath, „MABEL, a new robotic bipedal walker and runner,“ in *American Control Conference, 2009. ACC '09.*, 2009, pp. 2030–2036. DOI: 10.1109/ACC.2009.5160550.

[15] L. Grègoire, H. E. Veeger, P. A Huijing, and G. J. van Ingen Schenau, „Role of mono- and biarticular muscles in explosive movements,“ *International Journal of Sports Medicine*, vol. 5, 301–305, 1984.

[16] M. Günther and R. Blickhan, „Joint stiffness of the ankle and the knee in running,“ *Journal of biomechanics*, vol. 35, no. 11, pp. 1459–1474, 2002.

[17] I. Ha, Y. Tamura, H. Asama, J. Han, and D. Hong, „Development of open humanoid platform DARwIn-OP,“ in *SICE Annual Conference (SICE), 2011 Proceedings of*, 2011, pp. 2178–2181.

[18] T. Hemker, „Derivative free surrogate optimization for mixed-integer nonlinear black box problems in engineering,“ http://tuprints.ulb.tu-darmstadt.de/2162/, PhD thesis, Technische Universität Darmstadt, 2008. [Online]. Available: http://tuprints.ulb.tu-darmstadt.de/2162/.

[19] T. Hemker, H. Sakamoto, M. Stelzer, and O. von Stryk, „Efficient walking speed optimization of a humanoid robot,“ *International Journal of Robotics Research*, vol. 28, no. 2, pp. 303 –314, 2009. DOI: 10.1177/0278364908095171. [Online]. Available: http://dx.doi.org/10.1177/0278364908095171.

[20] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, „The development of Honda humanoid robot,“ in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, IEEE, vol. 2, 1998, pp. 1321–1326. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=677288.

[21] M. Hirose and K. Ogawa, „Honda humanoid robots development,“ *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 11–19, 2007. [Online]. Available: http://rsta.royalsocietypublishing.org/content/365/1850/11.short.

[22] K. Hosoda, Y. Sakaguchi, H. Takayama, and T. Takuma, „Pneumatic-driven jumping robot with anthropomorphic muscular skeleton structure,“ English, *Autonomous Robots*, vol. 28, no. 3, pp. 307–316, 2010, ISSN: 0929-5593. DOI: 10.1007/s10514-009-9171-6. [Online]. Available: http://dx.doi.org/10.1007/s10514-009-9171-6.

[23] K. Hosoda, T. Takuma, A. Nakamoto, and S. Hayashi, „Biped robot design powered by antagonistic pneumatic actuators for multi-modal locomotion,“ *Robotics and Autonomous Systems*, vol. 56, no. 1, pp. 46–53, 2008.

[24] J. D. Hunter, „Matplotlib: a 2D graphics environment,“ *Computing in science and engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[25] *IgH EtherCAT Master for Linux*. [Online]. Available: http://www.etherlab.org/en/ethercat/ (visited on Jul. 21, 2015).

[26] G. J. van Ingen Schenau, C. A. Pratt, and J. M. Macpherson, „Differential use and control of mono- and biarticular muscles,“ *Human Movement Science*, vol. 13, no. 3, pp. 495–517, 1994.

[27] *ISO/IEC 14882:2014 – Information technology – Programming languages – C++*, 2014.

[28] R. Jacobs, M. F. Bobbert, and G. J. van Ingen Schenau, „Mechanical output from individual muscles during explosive leg extensions: the role of biarticular muscles,“ *Journal of Biomechanics*, vol. 29, no. 4, pp. 513–523, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0021929095000674.

[29] S. Kajita, T. Nagasaki, K. Kaneko, K. Yokoi, and K. Tanie, „A running controller of humanoid biped HRP-2LR,“ in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, pp. 616–622. DOI: 10.1109/ROBOT.2005.1570186.

[30] K. Kaneko, F. Kanehiro, M. Morisawa, K. Akachi, G. Miyamori, A. Hayashi, and N. Kanehira, „Humanoid robot HRP-4-humanoid robotics platform with lightweight and slim body,“ in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, IEEE, 2011, pp. 4400–4407. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6094465.

[31] S. Kohlbrecher, A. Romay, A. Stumpf, A. Gupta, O. von Stryk, F. Bacim, D. Bowman, A. Goins, R. Balasubramanian, and D. Conner, „Human-robot teaming for rescue missions: team ViGIR´s approach to the 2013 DARPA Robotics Challenge trials,“ *Journal of Field Robotics*, vol. 32, no. 3, pp. 352–377, 2015, First published online 4 Dec 2014. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/rob.21558/full.

[32] A. M. Law, W. D. Kelton, and W. D. Kelton, *Simulation modeling and analysis*. McGraw-Hill New York, 1991, vol. 2.

[33] T. Lens, J. Kunz, C. Trommer, A. Karguth, and O. von Stryk, „BioRob-Arm: a quickly deployable and intrinsically safe, light- weight robot arm for service robotics applications,“ in *41st International Symposium on Robotics (ISR 2010) / 6th German Conference on Robotics (ROBOTIK 2010)*, Munich, Germany, 2010, pp. 905–910.

[34] T. Lens, K. Radkhah, and O. Von Stryk, „Simulation of dynamics and realistic contact forces for manipulators and legged robots with high joint elasticity,“ in *Advanced Robotics (ICAR), 2011 15th International Conference on*, 2011, pp. 34–41. DOI: 10.1109/ICAR.2011.6088619.

[35] H. ok Lim and A. Takanishi, „Biped walking robots created at Waseda University: WL and WABIAN family," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, 49–64, 2007. DOI: 10.1098/rsta.2006.1920. [Online]. Available: http://rsta.royalsocietypublishing.org/content/365/1850/49.abstract.

[36] *Linux preemt real-time kernel*. [Online]. Available: https://rt.wiki.kernel.org/ (visited on Jul. 21, 2015).

[37] S. Lipfert, „Kinematic and dynamic similarities between walking and running," PhD thesis, Friedrich-Schiller-Universität, Jena, Germany, 2009.

[38] X. Liu, A. Rosendo, M. Shimizu, and K. Hosoda, „Improving hopping stability of a biped by muscular stretch reflex," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, 2014, pp. 658–663. DOI: 10.1109/HUMANOIDS.2014.7041433.

[39] K. Loffler, M. Gienger, F. Pfeiffer, and H. Ulbrich, „Sensors and control concept of a biped robot," *Industrial Electronics, IEEE Transactions on*, vol. 51, no. 5, pp. 972–980, 2004, ISSN: 0278-0046. DOI: 10.1109/TIE.2004.834948.

[40] A. Martinez and E. Fernández, *Learning ROS for robotics programming*. Packt Publishing Ltd, 2013.

[41] *MATLAB - The Language of Technical Computing*. [Online]. Available: http://www.mathworks.com/products/matlab/ (visited on Jul. 21, 2015).

[42] *MatPlotLib - Plotting library for Python*. [Online]. Available: http://matplotlib.org/ (visited on Jul. 21, 2015).

[43] H.-M. Maus, S. W. Lipfert, M. Gross, J. Rummel, and A. Seyfarth, „Upright human gait did not provide a major mechanical challenge for our ancestors.," eng, *Nat Commun*, vol. 1, p. 70, 2010. DOI: 10.1038/ncomms1073. [Online]. Available: http://dx.doi.org/10.1038/ncomms1073.

[44] T. McGeer, „Passive dynamic walking," *the international journal of robotics research*, vol. 9, no. 2, pp. 62–82, 1990. [Online]. Available: http://ijr.sagepub.com/content/9/2/62.short.

[45] A. Mohammadinejad, M. Sharbafi, C. Rode, and A. Seyfarth, „Role of bi-articular muscles during swing phase of walking," in *Dynamic Walking 2013*, 2013.

[46] D. Nguyen-Tuong, J. Peters, M. Seeger, and B. Schölkopf, „Learning inverse dynamics: a comparison," in *European Symposium on Artificial Neural Networks*, 2008.

[47] R. Niiyama and Y. Kuniyoshi, „Design of a musculoskeletal Athlete robot: a biomechanical approach," in *Mobile Robitcs: Solutions and Challenges, Proc. of the 12th Int. Conf. on Climbing and Walking Robots (CLAWAR 2009)*, World Scientific Publishing, 2009, pp. 173–180, ISBN: 978-981-4291-26-2.

[48] R. Niiyama, S. Nishikawa, and Y. Kuniyoshi, „Athlete robot with applied human muscle activation patterns for bipedal running," in *Proc. IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids 2010)*, Nashville, Tennessee USA, 2010, pp. 498–503.

[49] *Orocos real-time toolkit (RTT).* [Online]. Available: http://www.orocos.org/rtt (visited on Jul. 21, 2015).

[50] H.-W. Park, A. Ramezani, and J. Grizzle, „A finite-state machine for accommodating unexpected large ground-height variations in bipedal robot walking," *Robotics, IEEE Transactions on*, vol. 29, no. 2, pp. 331–345, 2013. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6399609.

[51] S. Petters, D. Thomas, and O. Von Stryk, „Roboframe - a modular software framework for lightweight autonomous robots," in *Proc. Workshop on Measures and Procedures for the Evaluation of Robot Architectures and Middleware of the 2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007.

[52] G. A. Pratt and M. M. Williamson, „Series elastic actuators," in *IEEE International Workshop on Intelligent Robots and Systems*, 1995, 399–406.

[53] J. Pratt and B. Krupp, „Design of a bipedal walking robot," in *SPIE defense and security symposium*, International Society for Optics and Photonics, 2008, 69621F–69621F.

[54] J. Pratt and G. Pratt, „Intuitive control of a planar bipedal walking robot," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, IEEE, vol. 3, 1998, pp. 2014–2021. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=680611.

[55] *PyQt - Python bindings for Qt.* [Online]. Available: http://www.riverbankcomputing.com/software/pyqt/intro (visited on Jul. 21, 2015).

[56] *Python programming language.* [Online]. Available: http://www.python.org/ (visited on Jul. 21, 2015).

[57] *Qt application framework.* [Online]. Available: http://qt-project.org/ (visited on Jul. 21, 2015).

[58] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, „ROS: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, 2009, p. 5.

[59] K. Radkhah, T. Lens, and O. von Stryk, „Detailed dynamics modeling of BioBiped´s monoarticular and biarticular tendon-driven actuation system," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012, pp. 4243–4250.

[60] K. Radkhah, C. Maufroy, M. Maus, D. Scholz, A. Seyfarth, and O. von Stryk, „Concept and design of the BioBiped1 robot for human-like walking and running," *International Journal of Humanoid Robotics*, vol. 8, no. 3, pp. 439–458, 2011. DOI: 10.1142/S0219843611002587.

[61] K. Radkhah, „Advancing musculoskeletal robot design for dynamic and energy-efficient bipedal locomotion," http://tuprints.ulb.tu-darmstadt.de/3934/, PhD thesis, TU Darmstadt, Darmstadt, 2014. [Online]. Available: http://tuprints.ulb.tu-darmstadt.de/3934/.

[62] K. Radkhah and O. von Stryk, „Actuation requirements for hopping and running of the musculoskeletal robot BioBiped1," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, IEEE, 2011, pp. 4811–4818. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6094876.

[63] M. H. Raibert, *Legged robots that balance*. The MIT Press, 1986.

[64] S. Rapoport, J. Mizrahi, E. Kimmel, O. Verbitsky, and E. Isakov, „Constant and variable stiffness and damping of the leg joints in human hopping," *Journal of biomechanical engineering*, vol. 125, no. 4, pp. 507–514, 2003.

[65] *Robot operating system (ROS)*. [Online]. Available: http://www.ros.org/ (visited on Jul. 21, 2015).

[66] I. Rodriguez, A Astigarraga, E Jauregi, T Ruiz, and E Lazkano, „Humanizing NAO robot teleoperation using ROS," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, IEEE, 2014, pp. 179–186.

[67] *RTAI - the RealTime Application Interface for Linux*. [Online]. Available: https://www.rtai.org/ (visited on Jul. 21, 2015).

[68] M. Schilling, T. Hoinville, J. Schmitz, and H. Cruse, „Walknet, a bio-inspired controller for hexapod walking," *Biological cybernetics*, vol. 107, no. 4, pp. 397–419, 2013.

[69] D. Scholz, S. Kurowski, K. Radkhah, and O. von Stryk, „Bio-inspired motion control of the musculoskeletal BioBiped1 robot based on a learned inverse dynamics model," in *Proc. 11th IEEE-RAS Int. Conf. on Humanoid Robots (HUMANOIDS)*, Bled, Slovenia, 2011.

[70] D. Scholz, C. Maufroy, S. Kurowski, K. Radkhah, O. von Stryk, and A. Seyfarth, „Simulation and experimental evaluation of the contribution of biarticular gastrocnemius structure to joint synchronization in human-inspired three-segmented elastic legs," in *3rd Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR)*, 2012, pp. 251–260.

[71] D. Scholz and O. von Stryk, „Efficient design parameter optimization for musculoskeletal bipedal robots combining simulated and hardware-in-the-loop experiments," in *Proc. 15th IEEE-RAS Int. Conf. on Humanoid Robots, 2015*.

[72] M. Schwarz, M. Schreiber, S. Schueller, M. Missura, and S. Behnke, „NimbRo-OP humanoid teensize open platform," in *Proceedings of 7th Workshop on Humanoid Soccer Robots. IEEE-RAS International Conference on Humanoid Robots*, 2012.

[73] A. Seyfarth, M. Guenther, and R. Blickhan, „Stable operation of an elastic three-segment leg," *Biological Cybernetics*, vol. 84, no. 5, 365–382, 2001.

[74] A. Seyfarth, K. Radkhah, and O. von Stryk, „Soft robotics - transferring theory to application," in *Soft Robotics - Transferring Theory to Application*, A. Verl, A. Albu-Schäffer, O. Brock, and A. Raatz, Eds. Springer Verlag, 2015, ch. Concepts of Softness for Legged Locomotion and their Assessment, pp. 120–133.

[75] A. Seyfarth, R. Tausch, M. Stelzer, F. Iida, A. Karguth, and O. von Stryk, „Towards bipedal jogging as a natural result for optimizing walking speed for passively compliant three-segmented legs,“ *The International Journal of Robotics Research*, vol. 28, no. 2, 257–265, 2009.

[76] S. Shigemi, Y. Kawaguchi, T. Yoshiike, K. Kawabe, and N. Ogawa, „Development of new ASIMO,“ *HONDA R AND D TECHNICAL REVIEW*, vol. 18, no. 1, p. 38, 2006. [Online]. Available: http://sciencelinks.jp/j-east/article/200609/000020060906A0236517.php.

[77] *Simulink - Simulation and Model-Based Design*. [Online]. Available: http://www.mathworks.com/products/simulink/ (visited on Jul. 21, 2015).

[78] B. Stroustrup, *The C++ Programming Language*. Addison-Wesley, 2013.

[79] M. Summerfield, *Rapid GUI programming with Python and Qt: The definitive guide to PyQt programming*. Pearson Education, 2007.

[80] D. Thomas and D. Scholz, *rqt - GUI framework for ROS*, 2011. [Online]. Available: http://wiki.ros.org/rqt (visited on Jul. 21, 2015).

[81] D. Thomas, D. Scholz, S. Templer, and O. von Stryk, „Sophisticated offline analysis of teams of autonomous mobile robots,“ in *Proc. 5th Workshop on Humanoid Soccer Robots at the 2010 IEEE-RAS Int. Conf. on Humanoid Robots*, Nashville, TN, 2010.

[82] G. Van Rossum *et al.*, „Python programming language.,“ in *USENIX Annual Technical Conference*, vol. 41, 2007.

[83] A. Voronov, „The roles of monoarticular and biarticular muscles of the lower limbs in terrestrial locomotion,“ *Human Physiology*, vol. 30, no. 4, pp. 476–484, 2004. [Online]. Available: http://link.springer.com/article/10.1023/B:HUMP.0000036345.33099.4f.

[84] M. Vukobratovic and B. Borovac, „Zero-moment point: thirty five years of its life,“ *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0219843604000083.

[85] T. Wieland, *C++-Entwicklung mit Linux*. dpunkt-Verlag Heidelberg, 2003.

[86] *Xenomai - Real-time framework for Linux*. [Online]. Available: http://xenomai.org (visited on Jul. 21, 2015).

[87] S.-J. Yi, S. McGill, L. Vadakedathu, Q. He, I. Ha, J. Han, H. Song, M. Rouleau, D. Hong, and D. Lee, „THOR-OP humanoid robot for DARPA Robotics Challenge trials 2013,“ in *Ubiquitous Robots and Ambient Intelligence (URAI), 2014 11th International Conference on*, 2014, pp. 359–363. DOI: 10.1109/URAI.2014.7057369.

[88] F. E. Zajac, „Muscle coordination of movement: a perspective,“ *Journal of Biomechanics*, vol. 26, pp. 109–124, 1993.

Own Publications

Journal Papers

K. Radkhah, C. Maufroy, M. Maus, D. Scholz, A. Seyfarth, and O. von Stryk, „Concept and design of the BioBiped1 robot for human-like walking and running," *International Journal of Humanoid Robotics*, vol. 8, no. 3, pp. 439–458, 2011. DOI: 10.1142/S0219843611002587.

Conference Papers

D. Scholz and O. von Stryk, „Efficient design parameter optimization for musculoskeletal bipedal robots combining simulated and hardware-in-the-loop experiments," in *Proc. 15th IEEE-RAS Int. Conf. on Humanoid Robots, 2015*.

D. Scholz, C. Maufroy, S. Kurowski, K. Radkhah, O. von Stryk, and A. Seyfarth, „Simulation and experimental evaluation of the contribution of biarticular gastrocnemius structure to joint synchronization in human-inspired three-segmented elastic legs," in *3rd Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR)*, 2012, pp. 251–260.

D. Scholz, S. Kurowski, K. Radkhah, and O. von Stryk, „Bio-inspired motion control of the musculoskeletal BioBiped1 robot based on a learned inverse dynamics model," in *Proc. 11th IEEE-RAS Int. Conf. on Humanoid Robots (HUMANOIDS)*, Bled, Slovenia, 2011.

C. Maufroy, H.-M. Maus, K. Radkhah, D. Scholz, O. von Stryk, and A. Seyfarth, „Dynamic leg function of the BioBiped humanoid robot," in *Proc. 5th Int. Symposium on Adaptive Motion of Animals and Machines (AMAM)*, Osaka, Japan, 2011.

K. Radkhah, D. Scholz, A. Anjorin, M. Rath, and O. von Stryk, „Simple yet effective technique for robust real-time instability detection for humanoid robots using minimal sensor input," in *13th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR)*, Nagoya, Japan, 2010, pp. 680–689.

K. Radkhah, M. Maus, D. Scholz, A. Seyfarth, and O. von Stryk, „Towards human-like bipedal locomotion with three-segmented elastic legs," in *41st International Symposium on Robotics (ISR)/ 6th German Conference on Robotics (ROBOTIK)*, Munich, Germany, 2010, pp. 696–703.

Workshop Papers

D. Thomas, D. Scholz, S. Templer, and O. von Stryk, „Sophisticated offline analysis of teams of autonomous mobile robots," in *Proc. 5th Workshop on Humanoid Soccer Robots at the 2010 IEEE-RAS Int. Conf. on Humanoid Robots*, Nashville, TN, 2010.

C. Maufroy, M. Maus, K. Radkhah, D. Scholz, A. Seyfarth, and O. von Stryk, „First results for the BioBiped1 robot designed towards human-like walking and running,“ in *Workshop on Biomechanical Simulation of Humans and Bio-Inspired Humanoids, Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, Darmstadt, Germany, 2010.

D. Scholz, M. Friedmann, and O. von Stryk, „Fast, robust and versatile humanoid robot locomotion with minimal sensor input,“ in *Workshop on Humanoid Soccer Robots at the IEEE-RAS International Conference on Humanoid Robots*, 2009.

Open Source Code

D. Thomas and D. Scholz, *rqt - GUI framework for ROS*, 2011. [Online]. Available: `http://wiki.ros.org/rqt` (visited on Jul. 21, 2015).

Wissenschaftlicher Werdegang[1]

2000	Allgemeine Hochschulreife
2000 – 2003	Studium der Informatik, Johann Wolfgang Goethe-Universität Frankfurt
2003	Studium der Informatik, Bond University, Gold Coast, Australien
2004 – 2008	Studium der Informatik, Technische Universität Darmstadt
2008	Diplom in Informatik
2009 – 2014	Wissenschaftlicher Mitarbeiter, Technische Universität Darmstadt DFG-Projekt STR 533/7-1: Elastische, bionisch inspirierte, zweibeinige Roboter (BioBiped)
2009 – 2015	Doktorand, Fachgebiet Simulation, Systemoptimierung und Robotik, Fachbereich Informatik, Technische Universität Darmstadt

Erklärung[2]

Hiermit erkläre ich, dass ich die vorliegende Arbeit, mit Ausnahme der ausdrücklich genannten Hilfsmittel, selbständig verfasst habe.

[1] gemäß § 20 Abs. 3 der Promotionsordnung der TU Darmstadt
[2] gemäß § 9 Abs. 1 der Promotionsordnung der TU Darmstadt